

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Regulação Automática em Ambiente Metro-Ferroviário

João Pedro Vieira Peixoto

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Jorge Pinho de Sousa

Orientador na Empresa: Engenheiro José Manuel Sousa

29 de Julho de 2013

Resumo

A qualidade de serviço numa rede de transporte metro-ferroviário cobre múltiplos fatores, desde: conforto, segurança, pontualidade, rápida resposta a incidentes, entre outros. Estes dois últimos serão aqueles que mais sofrerão pela má gestão da rede em tempo real, ou pela tomada de decisões erradas realizadas pelas pessoas que têm de supervisionar os veículos em funcionamento e monitorizar os respetivos desvios entre os horários planeados e o real.

Este relatório de dissertação resulta de uma proposta da empresa EFACEC para estudar os Sistemas de Apoio à Regulação de Tráfego Metro-Ferroviário. Isto é, sistemas que têm como principal função auxiliar os controladores do tráfego a tomar decisões sobre os veículos de forma a reduzir ao máximo o impacto dos atrasos, ou então, que permita tomar ações em caso de falhas.

Este trabalho teve três fases distintas. Na primeira fase foi feita uma análise pormenorizada ao sistema que a EFACEC possui e foram identificados pontos onde este poderá ser melhorado. Numa segunda fase, foi escolhida uma função existente nesse sistema, o deslizamento, que tem como objetivo recuperar os horários de um atraso, tendo sido proposta uma nova forma de o fazer. Numa terceira fase foi desenvolvido o protótipo da aplicação informática de apoio à decisão, em WPF (XAML+C#), que tem como linhas de desenvolvimento os pontos identificados na primeira fase, e tendo como objetivo validar os algoritmos propostos.

No final deste trabalho, foram feitas alguns testes de forma a validar o correto funcionamento do protótipo desenvolvido. Para tal, foram utilizados horários e atrasos fictícios, tendo sido estes recuperados pelo sistema da forma pretendida.

Abstract

The quality of service in metro transport networks covers multiple factors: comfort, safety, punctuality, quick response to incidents, among others. These last two are the ones who will suffer for the bad management of the network in real time, or by making wrong decisions by people who have to supervise and monitor deviations between planned and actual schedules of vehicles in operation.

This report results from a dissertation proposal to study the Traffic Regulation Support Systems, made by a company, named EFACEC. This is a decision support system that has as main function auxiliary traffic controllers, to make decisions about the vehicles to reduce the impact of delays in planned schedules, or to allow them taking actions in case of failures.

The study had three distinct phases. In the first phase we made a detailed analysis of the system that EFACEC has and identified points where it can be improved. In the second phase an existing function has been selected from this system, slip, which aims to recover from a time delays, and has been proposed a new way of doing so. In a third step we developed a prototype implementation in WPF (XAML + C#), whose development lines points were identified in the first phase and the goal is validate all the proposed algorithms.

At the end of this work, some tests were made to validate the correct operation of the prototype. We used fictitious schedules and delays, and the system was capable of recover that delays as intended.

Agradecimentos

Em primeiro lugar, quero agradecer à pessoa que me fez a ser o quem sou, que me deu uma educação exemplar e me permitiu estar hoje onde estou. Obrigado Mãe, és e serás sempre a melhor!

Quero agradecer também a toda a minha família, avós, tios e primos, pois sempre acreditaram e me deram apoio para lutar pelo que queria.

Um obrigado especial à minha namorada, que foi capaz de aturar as minhas loucuras nos últimos meses, e que me deu sempre a força necessária.

Em relação ao desenvolvimento do trabalho quero agradecer aos meus dois orientadores, Professor Doutor Jorge Pinho de Sousa, e Engenheiro José Manuel Sousa. Ao professor, um obrigado pelos excelentes conselhos e pela forma pragmática com que me tentou fazer ver os problema. Espero ter captado um pouco desse pragmatismo. Ao Engenheiro José Manuel Sousa agradeço a disponibilidade e em especial a liberdade com que me permitiu desenvolver este projeto.

Por fim, agradeço a todos os colegas da EFACEC, que nos últimos 4 meses partilharam os seus dias comigo, em especial ao Carlos Faria, Luís Conceição, Daniel Rocha e Nuno Rocha.

João Pedro Peixoto

*“The greater our knowledge increases,
the greater our ignorance unfolds.”*

John F. Kennedy

Conteúdo

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 1.1 | Motivação | 1 |
| 1.2 | Metodologia | 2 |
| 1.3 | Objetivos | 2 |
| 1.4 | Estrutura do documento | 3 |
| 2 | Revisão bibliográfica | 5 |
| 2.1 | Conceitos base | 5 |
| 2.1.1 | Terminologia | 5 |
| 2.1.2 | Diagramas de viagem | 7 |
| 2.2 | Estado da arte | 9 |
| 2.2.1 | Sistemas de Apoio à Decisão | 9 |
| 2.2.2 | Sistema de Regulação ou Traffic Regulation Support System | 10 |
| 2.2.3 | A arquitetura do Sistema de Regulação | 11 |
| 2.2.4 | Algoritmos de regulação | 12 |
| 3 | Módulo de Regulação da EFACEC | 15 |
| 3.1 | Solução chave-na-mão | 15 |
| 3.2 | Sistema de Apoio à Exploração | 16 |
| 3.3 | Módulo de regulação | 18 |
| 3.3.1 | Manobras de regulação | 20 |
| 3.3.2 | Algoritmos de otimização | 21 |
| 3.3.3 | Limitações da plataforma | 21 |
| 4 | Caso de estudo | 23 |
| 4.1 | Problema a tratar | 23 |
| 4.2 | Notação | 24 |
| 4.3 | Restrições | 25 |
| 4.4 | Recuperação de atrasos | 26 |
| 5 | Método de resolução | 29 |
| 5.1 | Introdução | 29 |
| 5.2 | Classificação de atrasos | 30 |
| 5.3 | Algoritmos | 31 |
| 5.3.1 | Coefficiente de recuperação | 31 |
| 5.3.2 | Algoritmo 1 | 33 |
| 5.3.3 | Algoritmo 2 | 35 |

| | | |
|----------|---|-----------|
| 6 | O protótipo | 37 |
| 6.1 | Descrição geral | 37 |
| 6.1.1 | Introdução | 37 |
| 6.1.2 | As linguagens e os ambientes de desenvolvimento | 38 |
| 6.1.3 | Model View ViewModel | 39 |
| 6.2 | Descrição do interface | 40 |
| 6.3 | Resultados | 42 |
| 7 | Conclusões | 47 |
| 7.1 | Conclusões sobre o trabalho realizado | 47 |
| 7.2 | Conclusões sobre a experiência empresarial | 49 |
| 7.3 | Trabalhos futuros | 49 |
| | Referências | 51 |

Lista de Figuras

| | | |
|-----|--|----|
| 2.1 | Figura ilustrativa de duas vias, um veículo e uma estação | 6 |
| 2.2 | Exemplo de um diagrama de viagem com 3 serviços | 7 |
| 2.3 | Diagrama de viagens com um cenário real | 8 |
| 2.4 | Diagrama de viagem com conflitos | 8 |
| 2.5 | Arquitetura geral de um SAD | 10 |
| 2.6 | Arquitetura geral de um TRSS, ou sistema de apoio à regulação de tráfego | 12 |
| 3.1 | Elementos constituintes da solução chave-na-mão da EFACEC [1] | 16 |
| 3.2 | Arquitetura geral do TimeKeeper | 17 |
| 3.3 | Janela principal do SAE da EFACEC | 18 |
| 3.4 | Janela principal do módulo de regulação da EFACEC | 19 |
| 5.1 | Representação gráfica de atraso Tipo 1 | 31 |
| 5.2 | Representação gráfica de atraso Tipo 2 | 32 |
| 5.3 | Algoritmo 1 | 34 |
| 5.4 | Algoritmo 2 | 36 |
| 6.1 | Arquitetura geral do padrão de desenvolvimento <i>Model View ViewModel</i> [2] | 40 |
| 6.2 | Aspeto geral do interface do protótipo desenvolvido | 41 |
| 6.3 | Tabela de navegação dos horários | 42 |
| 6.4 | Figura ilustrativa da forma de funcionamento da aplicação | 43 |
| 6.5 | Resultado do deslizamento de um atraso de 3 minutos com coeficiente de recuperação mínimo | 43 |
| 6.6 | Resultado do deslizamento de um atraso de 3 minutos com coeficiente de recuperação máximo | 44 |
| 6.7 | Resultado do deslizamento de um atraso de 3 minutos com coeficiente de recuperação médio | 44 |
| 6.8 | Resultado do deslizamento de um atraso de 10 minutos com coeficiente de recuperação máximo | 45 |
| 6.9 | Resultado do deslizamento de um atraso de 10 minutos com coeficiente de recuperação mínimo | 46 |

Lista de Tabelas

| | | |
|-----|--|----|
| 5.1 | Simulação da variação do coeficiente de recuperação em função do número de iterações, com um valor de entrada inicial de 0.4 | 33 |
|-----|--|----|

Abreviaturas e Símbolos

| | |
|------|---|
| AVL | <i>Automatic Vehicle Location</i> |
| AVM | <i>Automatic Vehicle Monitoring</i> |
| IDE | <i>Integrated Development Environment</i> |
| MVC | <i>Model View Controller</i> |
| MVVM | <i>Model View ViewModel</i> |
| PCC | Posto Central de Comando |
| SAD | Sistema de Apoio à Decisão |
| SAE | Sistema de Apoio à Exploração |
| TRSS | <i>Traffic Regulation Support System</i> |
| WPF | <i>Windows Presentation Foundation</i> |
| XAML | <i>eXtensible Application Markup Language</i> |

Capítulo 1

Introdução

Este documento foi desenvolvido no âmbito da unidade curricular de Dissertação do ano de 2012/2013, tendo como propósito relatar todo o trabalho desenvolvido ao longo do projeto desenvolvido.

Este relatório de dissertação resultou de uma proposta de uma empresa, mais concretamente do Grupo EFACEC, e teve como ponto de partida a problemática da regulação em ambiente metro-ferroviário. Isto é, um sistema de apoio à decisão que tem como principal função auxiliar os controladores do tráfego, a tomar decisões sobre os veículos com o objetivo de manter a circulação o mais semelhante ao planeado, ou então que permita tomar ações em caso de falhas.

É importante clarificar, desde já, o sentido do termo "regulação". Neste contexto, a regulação é o processo de supervisão de uma rede de transportes, podendo ser estes ferroviários, metro-ferroviários, ou terrestres, tendo como principal meta garantir o cumprimento dos horários, ou responder a eventos imprevisíveis que de forma direta ou indireta prejudiquem a normal circulação dos veículos.

1.1 Motivação

Nos últimos anos tem sido cada vez maior a preocupação das companhias de redes de transporte em fornecer um serviço aos seus utilizadores com maior qualidade. Isto deve-se principalmente ao aumento da procura, existência de maior competição no setor e pela privatização de muitas destas companhias [3]. A qualidade de serviço nas redes de transporte cobre múltiplos fatores desde conforto, segurança, pontualidade, rápida resposta a incidentes, entre outros. Estes dois últimos serão aqueles que mais sofrerão pela uma má gestão da rede em tempo real, ou seja tomada de decisões erradas realizadas pelas pessoas que têm de supervisionar os veículos em funcionamento e monitorizar respetivos desvios entre os horários planeados e o real.

A pontualidade em conjunto com o tempo de espera são dos fatores mais importantes para a qualidade do serviço, na perspetiva do passageiro. No entanto, por muito que o planeamento seja bem realizado, existirão sempre falhas aleatórias, impossíveis de prever, que provocarão atrasos

nos veículos, resultando numa menor qualidade de serviço para o utilizador. É em pleno funcionamento de atividade das redes de transportes que é necessário tomar medidas com o objetivo de garantir que o impacto na rede seja o menor possível, sendo que estas decisões deverão ser tomadas num curto espaço de tempo.

Nos últimos anos, muitas das companhias que oferecem serviços de transporte coletivo instalaram ferramentas de deteção dos seus veículos em tempo real, fazendo com que seja sempre possível saber onde estes se encontram a todo o momento. Estas informações, entre outras, serão o suporte para os controladores do tráfego da rede poderem fazer reajustes no planeamento previsto.

O problema em questão não terá uma solução ótima, pois como se pode perceber facilmente existem múltiplos objetivos (qualidade de serviço, intervalo de tempo mínimo entre veículos, tempos de mudança condutor, tempos de rota definidos, etc.). A acrescentar a isto também estamos a falar de uma rede de veículos interligada, onde a alteração em um veículo poderá ter impacto em todos os outros. Além disso, trata-se de um problema combinatório, de muito difícil resolução [4]. Atendendo à restrição do tempo de ação é praticamente impossível para um humano encontrar uma solução para o problema. Por estes motivos, é necessário um software que permita apoiar o controlador nesta tarefa, permitindo que este em poucos segundos consiga tomar uma boa decisão. A estes sistemas dá-se o nome de SAD (Sistemas de Apoio à Decisão).

Este é um problema real, complexo, para o qual já existem soluções informáticas, como é o caso da solução da EFACEC que é utilizada em vários países do mundo, como por exemplo Argélia, Irlanda, Inglaterra e Dinamarca.

1.2 Metodologia

A metodologia utilizada para o desenvolvimento desta dissertação teve várias fases. Numa fase inicial, foi feito um levantamento bibliográfico sobre os fundamentos teóricos do problema, recorrendo a artigos científicos e dissertações. Numa fase posterior passou-se a análise da solução de software oferecida pela EFACEC, onde foi analisada toda a arquitetura onde esta solução está integrada e tentou-se identificar os pontos menos fortes da mesma. Em paralelo estudou-se as linguagens de programação C# e XAML, que foram utilizadas para a implementação do protótipo da aplicação.

1.3 Objetivos

Um dos objetivos desta dissertação é estudar uma arquitetura real de um sistema de apoio à regulação do tráfego e conseguir identificar pontos de melhoria do mesmo. Este estudo teve duas fases, numa fase inicial, foi feito um estudo mais teórico, e numa segunda fase, foi feita a análise ao sistema real da EFACEC.

O segundo objetivo passa por estudar uma das manobras mais utilizadas neste tipo de sistema, a manobra de deslizamento, que será mais à frente explicada em pormenor. Mas, de uma forma simples, o objetivo desta é encontrar uma forma para poder recuperar um atraso de um veículo.

O terceiro objetivo passou pela implementação de raiz de um sistema de apoio à decisão que implementasse a manobra previamente estudada. Este desenvolvimento teve algumas peculiaridades, pois tentou-se fazer um desenvolvimento a partir de conceitos base de engenharia de software, com a intenção de validar as vantagens de seguir um padrão de desenvolvimento quando se devolve software complexo.

1.4 Estrutura do documento

Este documento está dividido em 7 capítulos: Introdução, Revisão Bibliográfica, Módulo de Regulação da EFACEC, Caso de Estudo, Método de Resolução, Protótipo e Conclusões.

Neste primeiro capítulo é feita uma breve apresentação da temática, da motivação, metodologia utilizada e os principais objetivos da dissertação. A ideia será dar ao leitor uma breve noção do que foi abordado durante o trabalho.

O segundo capítulo, Revisão Bibliográfica, está dividido em duas partes, num primeiro momento será apresentada a terminologia adotada e é explicado a importância dos diagramas de viagem. Numa segunda parte é feito um breve levantamento da literatura sobre a temática. Este capítulo terá como função munir o leitor dos conceitos necessários para poder seguir o resto do documento.

No terceiro capítulo é exposta a plataforma de regulação da EFACEC, apresentando a solução comercializada pela empresa, passando pela a arquitetura da plataforma e em seguida é feita uma descrição extensiva do módulo de regulação existente. Depois são também mostradas algumas críticas à ferramenta.

O quarto capítulo apresenta a descrição do caso de estudo, define fronteiras e algumas simplificações, notação do problema, restrições, e explica de que forma os atrasos se podem recuperar. Este capítulo tem preponderância no projeto, pois além de nos mostrar o problema de uma forma menos subjetiva também nos tenta transmitir toda a dinâmica de uma rede de metros.

No quinto capítulo são descritos os métodos de resolução adotados para o caso de estudo. Neste capítulo são mostrados todos os algoritmos propostos e quais as intenções dos mesmos.

O sexto capítulo irá explicar de que forma foi desenvolvido o protótipo da aplicação, quais os pressupostos e as metas a alcançar. Não iremos entrar em pormenores técnicos de implementação mas iremos explicar o porquê e forma como tudo foi pensado e implementado. Não sendo um projeto de desenvolvimento de software, este também tem esta componente, tendo, por isso, sido dada alguma relevância ao assunto. Também pelo facto deste projeto ser desenvolvido numa empresa, onde uma boa estrutura de desenvolvimento poderá por em causa o sucesso do trabalho. No final deste capítulo são apresentados os resultados conseguidos.

O último capítulo é o das Conclusões. O objetivo é resumir tudo o aprendido e perceber qual foi o real valor do projeto. Por fim, também é feita referência a possíveis trabalhos futuros.

Capítulo 2

Revisão bibliográfica

Este capítulo terá como função familiarizar o leitor com a temática. Está dividido em duas partes. Na primeira parte é apresentado a terminologia utilizada ao longo do documento, e alguns conceitos chave para ser possível uma correta interpretação do mesmo. Na segunda parte é apresentado um apanhado geral da revisão bibliográfica realizada.

2.1 Conceitos base

2.1.1 Terminologia

Antes de entrar na temática propriamente dita é preciso definir alguma terminologia. É importante referir que a terminologia adotada foi baseada na terminologia utilizada pela própria empresa, podendo esta variar em outra literatura, ou outro contexto empresarial.

As definições apresentadas são chave para o bom entendimento do documento, por isso é aconselhado recorrer a este capítulo quando se sentir menos confiante quanto à interpretação que deve ser dada a uma certa expressão.

Rede de Metro – é o conjunto de todos os elementos que fazem parte de um sistema metro-ferroviário. Normalmente é constituído por: linha ou linhas, estações, ponto de mudança, pontos de viragem, terminais, depósito.

Linha - é um conjunto de vias, estações e outros elementos que ligam os dois extremos de cada via. Em jeito de nota, na realidade poderão não ser dois extremos de cada via, mas para o contexto do problema é importante pensar dessa forma quando nos referimos à linha.

Via – quando nos referimos a via, estamos a falar concretamente do material por onde o metro circula (carris), na imagem [2.1](#) a baixo podemos ver duas vias distintas.

Terminal - ponto de extremo das linhas.

Estação – Localização física que agrega um dado conjunto de plataformas.

Plataforma - Local onde se procede à entrada e saída de passageiros para os veículos circulantes num sentido de trânsito.

Headway – Diferença de tempo entre dois veículos que circulam na mesma via e que seguem um atrás do outro.



Figura 2.1: Figura ilustrativa de duas vias, um veículo e uma estação

Viagem – percurso feito por um veículo entre dois pontos distintos da rede, em maioria das situações estes pontos são os terminais da linha. No entanto poderão ser definidos outros pontos dependendo das necessidades ou adversidades da operação. A título de exemplo, uma viagem ligará o ponto “A” ao ponto “B”, ou seja somente é considerada a ida, sendo outra viagem, a volta.

Serviço – é um conjunto de viagens encadeadas entre si, ou seja, um conjunto de viagens ida e volta, entre dois pontos definidos. Normalmente são realizadas pela mesmo veículo. Mas, o importante é perceber que as viagens dentro dos serviços estão encadeadas entre si, por exemplo a viagem 2 só começará quando a viagem 1 terminar.

Secção – parte de uma via.

Trajetos / Rota – Isto é uma sequencia de secções da rede ligadas entre si, é a descrição física da rota que o veículo tem que fazer para ir de um ponto a outro ponto destino.

Horários – Lista das horas às quais os veículos passarão em determinado ponto na rede, usualmente estes pontos são paragens.

Serviços / Viagens / Horários Planeados – Quando se fala de serviços, viagens ou horários planeados estamos a fazer referência a todos aqueles que foram construídos numa fase prévia à da operação.

Posto Central de Comando (PCC) - Local onde é realizada a monitorização e supervisão das linhas.

Operador / Regulador / Controlador - Estes termos serão utilizados como sinónimos e referem-se às pessoas que estão no Posto Central de Comando a monitorizar e supervisionar a rede metro ferroviária.

2.1.2 Diagramas de viagem

Os operadores que têm a tarefa de supervisionar uma rede de metro têm duas ferramentas base para poderem fazer o seu trabalho: os horários e os diagramas de viagem ou diagramas espaço/tempo. Os horários já foram definidos anteriormente, como sendo uma lista com as horas às quais os veículos passam nas plataformas. No entanto, estes poderão ainda ter mais informação como por exemplo, as horas a que os veículos passarão em determinado local da via, que não uma estação.

Os diagramas de viagem serão talvez a ferramenta mais poderosa ao serviço dos operadores. Como é facilmente perceptível é uma tarefa complicada conseguir detetar conflitos, ou então possíveis situações de perturbação numa linha através dos seus horários. Uma lista de itens é bastante útil quando é necessário procurar algo específico, no entanto perde qualidade quando é necessário uma visão global do sistema. Pelo contrário, uma representação visual da informação, através de um gráfico, garante-nos uma maior capacidade de análise do sistema como um todo. Os diagramas de viagem, são uma representação do horário de uma forma gráfica, onde o eixo dos xx será o tempo, e o eixo do yy o espaço. Na figura 2.2 temos um exemplo ilustrativo de um diagrama de viagens simplificado, onde podemos ver 3 linhas de cores diferentes, sendo que cada uma delas representam serviços distintos. Como podemos observar os declives vão variando entre valores positivos e negativos, dando-nos a informação do sentido que os veículos circulam.

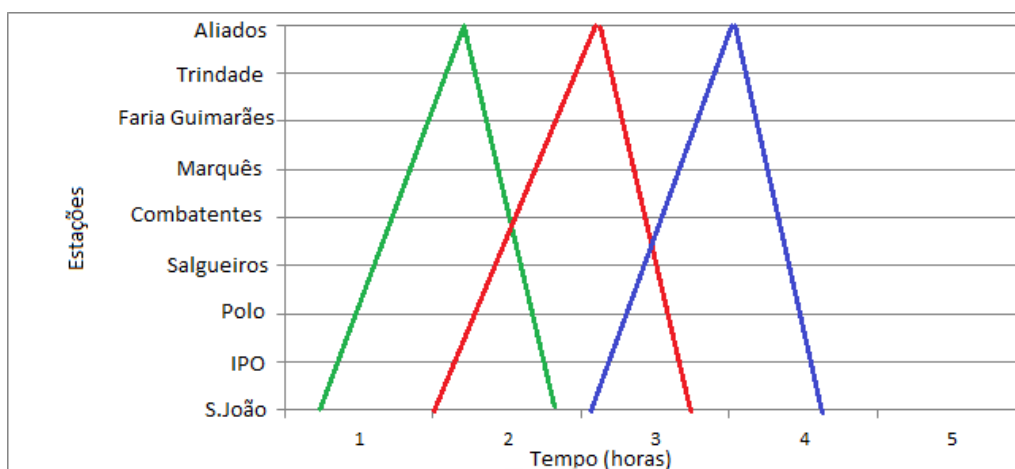


Figura 2.2: Exemplo de um diagrama de viagem com 3 serviços

Na figura 2.3, é apresentado um exemplo mais realista, retirado do software pertencente à EFACEC, sendo que é possível ver um grande número de linhas a cruzarem-se entre si. Neste caso concreto é importante referir que este cruzamento de linhas não indica qualquer tipo de conflito entre os veículos, pois estamos a falar de linhas onde existem 2 vias, sendo cada uma delas para a circulação em sentido único. No entanto é importante referir que esta interpretação dos diagramas estará dependente da configuração da linha em questão, é possível encontrar na literatura exemplos de problemas semelhantes onde o cruzamento de linhas com declives inversos indicará existência de conflito [5].

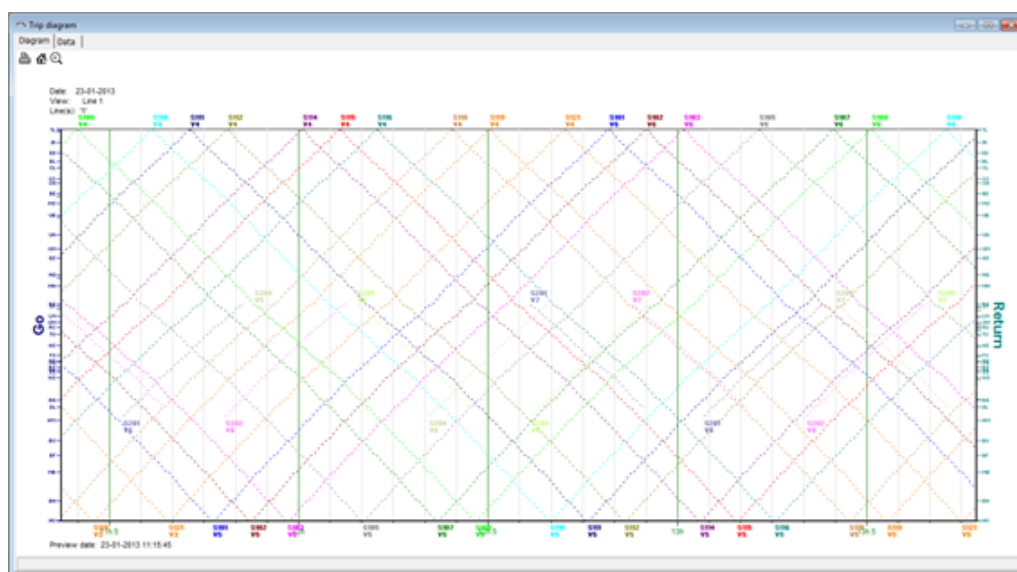


Figura 2.3: Diagrama de viagens com um cenário real

É então, de extrema relevância definir quais serão os conflitos que se poderão detetar através de um diagrama deste tipo numa configuração de metro com 2 vias de sentido único. Na figura 2.4, estão representados os conflitos possíveis de detetar através deste tipo de diagrama.

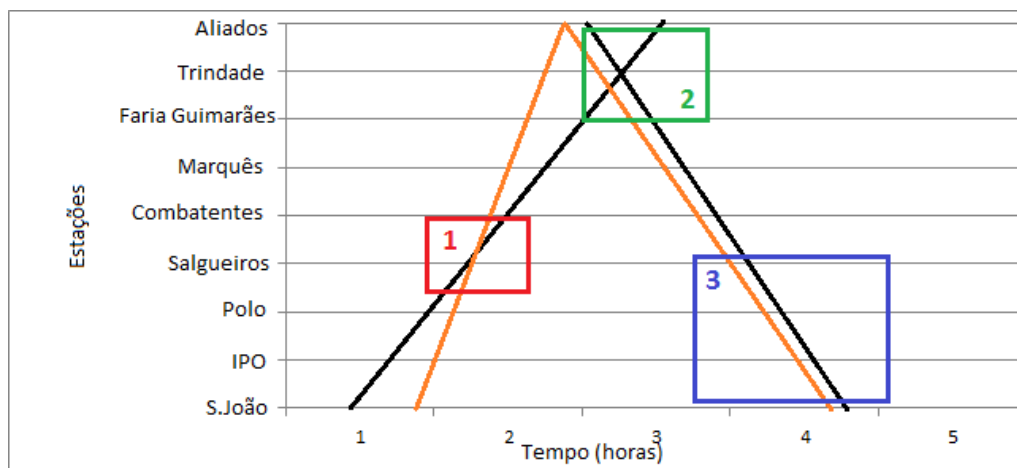


Figura 2.4: Diagrama de viagem com conflitos

Assim, temos:

1. No quadrado vermelho podemos ver um cruzamento de linhas de cores distintas com um declive com mesmo sinal. Este conflito diz-nos que havia um veículo a circular atrás de outro na mesma via e este ultrapassou-o. No entanto fisicamente isso é impossível pois não existe forma de ultrapassar um veículo que circula na mesma via.

2. Como foi dito acima, é completamente normal haver um cruzamento de linhas que tenham

declives de sinais diferentes, isso significa que estão a circular em sentidos apostos em vias distintas. No entanto, existe um caso excecional onde tal indicará um conflito, representado na imagem a verde, a linha que se cruza é da mesma cor, indicando que é o mesmo serviço em questão, tal indica que a segunda viagem começou antes da primeira ter terminado.

3. No quadrado azul vemos duas linhas com o mesmo sinal de declive praticamente juntas, isto querará indicar que existem dois veículos a circular na mesma via muito próximos um do outro. Na realidade poderá ser ou não um conflito, pois dependerá das restrições técnicas de cada linha, sendo que normalmente existe uma distância de segurança mínima para veículos que circulam uns atrás dos outros.

As três situações enumeradas são as mais importantes para o problema, pois são ou poderão indicar um conflito. No entanto há mais informação que se pode retirar deste tipo de diagrama, como por exemplo, é facilmente detetável ver se a regularidade dos serviços esta a ser feita de forma constante ou se pelo contrário existe diferenças grandes de tempo entre a passagem de um veículo num local e a passagem de outro. Essa informação é facilmente encontrada através da distância horizontal entre as linhas com o mesmo declive.

2.2 Estado da arte

2.2.1 Sistemas de Apoio à Decisão

Nos dias de hoje os SAD, são um dos domínios mais estudados no âmbito dos sistemas de informação. Burstein e Holsapple definem sistemas de apoio à decisão como: sistemas onde é representada e processada a informação com o intuito de melhorar as decisões tomadas (“system wich represent and process information for the purpose of improving decision making”), O’ Brien define estes sistemas como sistemas computacionais que fornecem informação de uma forma interativa para ajudar os gestores e profissionais de negócio durante o processo da tomada de decisão (“computer-based systems that provide interactive information support to managers and business professionals during the decision making process”) [6]. Existem várias definições para SAD, mas todas elas convergem para: sistemas computacionais que têm como objetivo facilitar a tomada de decisão de um problema complexo. Este tipo de ferramenta pode ser uma simples folha de cálculo, como poderosos softwares com grandes bases de dados e algoritmos complexos.

Os SAD tem todos uma arquitetura genérica, como o da figura 2.5, onde podemos perceber a interação entre os diferentes elementos do modelo [7].

A partir da arquitetura de um SAD podemos perceber que a solução proposta vai depender de como o SAD foi desenvolvido, ou seja, vai depender de como este foi construído e implementado. Desta forma, para um mesmo problema poderão existir múltiplas formas de o resolver. Recentemente têm emergido no campo do SAD a utilização de sistemas mais sofisticados, tais como: multiagente, lógica difusa, algoritmos genéticos, etc [4].

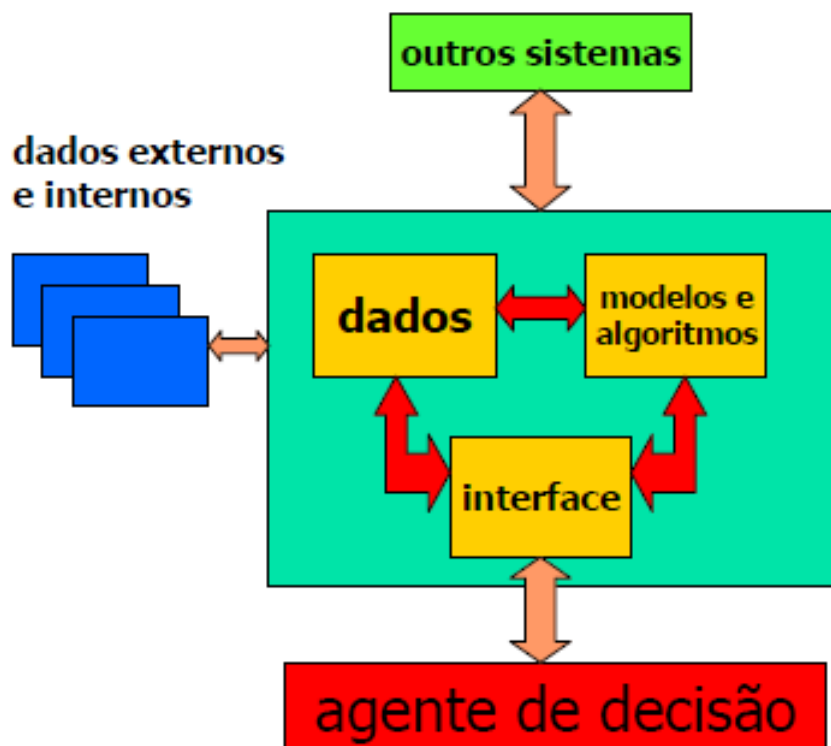


Figura 2.5: Arquitetura geral de um SAD

No campo das redes de transportes já há muito que os SAD são utilizados, sendo muitas vezes estes apelidados na literatura inglesa de Traffic Regulation Support Systems (TRSS), estes sistemas não são mais do que sistemas de apoio à regulação do tráfego. Estes caracterizam-se por ser sistemas de apoio à decisão em tempo real, que deverão prever o aparecimento de falhas ou incidentes no planeamento e reagir às mesmas. No entanto se não for possível a sua previsão, estes deverão ser capazes de identificá-las, analisá-las e avaliar o impacto destas na rede. No momento seguinte o sistema deverá propor soluções ao controlador e este deverá escolher que ações tomar [3]

2.2.2 Sistema de Regulação ou Traffic Regulation Support System

Nas redes de transporte urbanos os controladores estão localizados no PCC (Posto Central de Comando), e é a partir desse local que devem gerir toda a rede. Para desempenhar esta tarefa, estes precisam de saber qual a localização de cada veículo no momento e ainda se existe algum veículo que se encontra fora do planeado (atrasado, adiantado, bloqueado, acidentado, etc).

Nas redes modernas a localização dos veículos é feita através de sensores que enviam a informações para o PCC em tempo real. Esta recolha de informação é feita pelo AVM (Automatic Vehicle Monitoring System). O sistema AVM permite comparar as posições dos veículos com a

sua posição teórica, dada pelas tabelas de horários antes definidas, e ainda lançar alarmes para o controlador quando existem desvios.

A função do sistema AVM é processar *online* a informação básica acerca dos veículos, organizar esta mesma informação e apresentá-la ao controlador. O interface do mesmo facilita a compreensão da informação e permite obter mais detalhes sobre paragens ou veículos se o operador o requisitar.

No entanto os controladores são expostos a grandes quantidades de informação e mesmo com a ajuda do sistema AVM a sua tarefa é ainda bastante complicada. É ainda importante referir que estes controladores passam ainda 50% do seu tempo em comunicação com os condutores dos veículos. E por isso, para estes poderem tomar boas decisões devem [8]:

- ter uma ideia global de rede;
- fazer uma análise espaço-tempo do desvio;
- simular o efeito das ações na rede.

É então humanamente impossível desempenhar esta tarefa sem ferramentas de apoio, e é aqui que se encaixa o TRSS, ou sistema de regulação. Numa rede de transporte o TRSS terá a função de identificar e avaliar os desvios existentes, a partir dos dados provenientes do AVM. Este sistema deverá ser capaz de sugerir soluções, no entanto deve dar liberdade total ao controlador de poder encontrar outra solução e também ser capaz de simular o impacto das mesmas na rede.

2.2.3 A arquitetura do Sistema de Regulação

As redes de transporte metro ferroviário são sistemas marcados por repetição e periodicidade. No entanto, como foi explicado em cima existem sempre atrasos não previstos, muitas vezes provocados pelos próprios passageiros devido à variação de tempo que obrigam o veículo a estar parado na estação. Também existem incidentes que podem provocar atrasos nos veículos. Muitas vezes estes atrasos podem ser apenas segundos mas poderão ter impacto na rede, por exemplo em paris os metros circulam com uma diferença média entre veículos de 95 segundos, sendo que em hora de ponta este valor poderá rondar valores na ordem dos 20 segundos [9].

É então essencial possuir ferramentas de supervisão e monitoramento como o AVM e TRSS. Como foi dito antes, os sistemas de regulação são uma peça complementar ao sistema de AVM, permitindo aos controladores uma maior fiabilidade de operação.

Na figura 2.6 é possível perceber qual é a arquitetura geral do sistema: o módulo AVM recebe a informação da localização dos veículos e compara com a posição teórica que esta em base de dados, este por fim entrega a informação ao TRSS. O TRSS tem um interface que mostra as posições dos veículos e respetivos distúrbios detetados pelo AVM, a partir desses dados o operador fará a sua análise e decidirá se deve ou não atuar sobre a rede. Se o fizer, este dará uma ordem ao TRSS e escolherá o algoritmo que quer utilizar para construir uma solução. O algoritmo sendo o cérebro do sistema constrói a solução e entrega ao operador, que terá como tarefa decidir qual a opção a implementar.

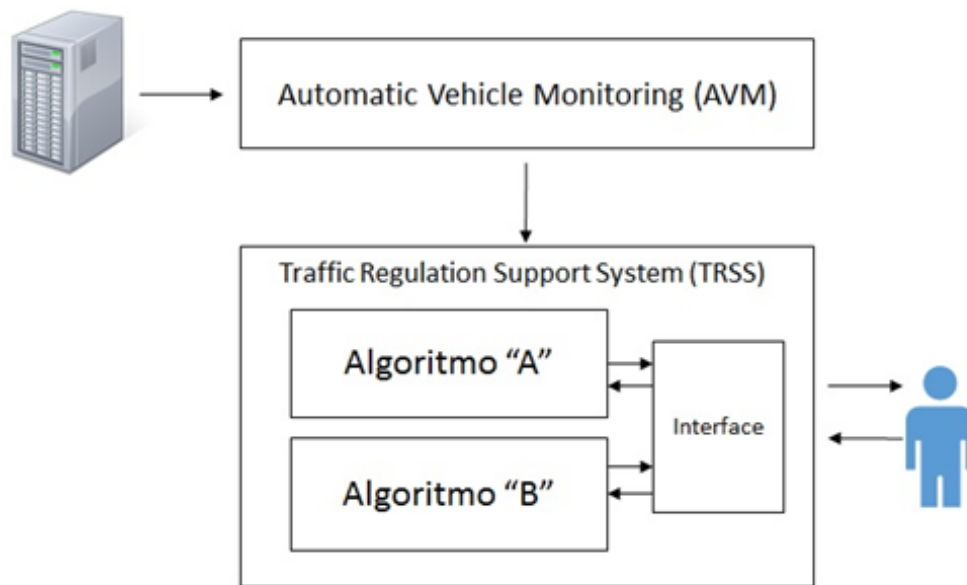


Figura 2.6: Arquitetura geral de um TRSS, ou sistema de apoio à regulação de tráfego

As grandes diferenças entre os vários TRSS propostos estão no algoritmo do sistema, e neste módulo que a investigação tem dedicado a atenção, pois tendo em consideração que o problema não tem uma solução ótima, é importante encontrar um algoritmo ou algoritmos que consigam encontrar soluções o mais próximo da solução ótima.

2.2.4 Algoritmos de regulação

Nesta secção referem-se os algoritmos que têm sido desenvolvidos para este tipo de sistema, começando pelos algoritmos típicos utilizados nas redes de transporte metro ferroviárias e por fim apresentando uma referência a algumas inovações, que se centram muito pela incorporação de algoritmos inteligentes no sistema.

Os algoritmos que tipicamente são utilizados neste tipo de sistema são: regulação por horário, regulação por intervalo e regulação mista.

Na regulação por horário, o algoritmo terá como objetivo olhar para cada veículo individualmente e procurar que este volte o mais rápido possível ao horário planeado. A ideia base deste algoritmo é tentar alterar os tempos de partida dos veículos e também o tipo de condução de cada veículo para que estes voltem ao horário previsto. Este tipo de algoritmo tem algumas limitações pois não tem em conta a rede de uma forma global e o impacto que a alteração de um veículo tem em outros veículos, no entanto em casos onde os atrasos são curtos e individuais poderá ter bons resultados [9].

Na regulação por intervalo, o sistema irá procurar uma solução tendo por base o intervalo de tempo entre os veículos, ignorando por completo os horários. Este tipo de algoritmo só deve ser

utilizado em situações em que existem vários veículos da rede que estão desviados do planeado [9].

Na regulação mista, é aplicada a vários veículos em simultâneo tendo como meta restabelecer o intervalo de tempo entre os veículos em questão, e em paralelo o horário dos mesmos [9].

Os métodos antes enumerados são os mais antigos e ainda muito utilizados neste tipo de sistemas. No entanto, nos últimos anos tem-se observado uma corrente que procura incorporar algoritmos cada vez mais sofisticados neste tipo de sistema.

Em 2008, Mohamed Mahmoud Ould Sidi, Slim Hammadi, Said Hayat and Pierce Borne, desenvolveram uma solução híbrida, baseado num sistema fuzzy para avaliar as soluções e um algoritmo evolutivo usando a teoria de Pareto para a construção das mesmas. Este sistema propõe diferentes alternativas ao operador e uma classificação das mesmas. Segundo os autores, o algoritmo evolutivo utilizado garante uma diversidade de soluções, ao mesmo tempo que integra no sistema fuzzy a capacidade de avaliar a solução de uma ótica mais subjetiva, baseado em experiências passadas[10].

Em 2010, é publicado um artigo por Flavien Balbo e Suzanne Pinson, que resulta da construção de um TRSS baseado num sistema multi-agente. Segundo os autores a utilização desta metodologia tem vantagens quando utilizada em sistemas de monitorização distribuídos geograficamente[11].

Em 2012, Selim Dundar e Ismail Sahin compara soluções obtidas por uns algoritmos genéticos, com as soluções obtidas através de uma rede neuronal. O objetivo do autor era comparar qual das duas abordagens responderia melhor quando existem situações conflituosas entre os diferentes veículos, para tal a rede neuronal foi treinada com dados recolhidos de casos anteriores onde os problemas foram solucionados pelos próprios controladores. Os autores concluem que o algoritmo genético construído terá melhores resultados que a rede neuronal desenvolvida[12].

Nos últimos tempos tem-se assistido a um aumento de propostas para TRSS baseadas em técnicas mais sofisticadas, além dos já referidos também é possível encontrar soluções baseados em: sistemas inteligentes, métodos heurísticos e meta-heurísticas (pesquisa tabu, *simulated annealing*) e também solução híbridas. No entanto, para o problema concreto do projeto não iremos utilizar nenhuma desses algoritmos. O caso de estudo tem restrições computacionais bastante restritas sendo preferível um algoritmo simples e rápido que permita construções de soluções em poucos segundos.

Capítulo 3

Módulo de Regulação da EFACEC

Neste capítulo apresenta-se de forma detalhada o sistema onde está inserido o sistema de apoio à regulação de tráfego da EFACEC, ou como é conhecido na empresa, o módulo de regulação. Inicialmente, é apresentada de uma forma sucinta o produto que a EFACEC oferece. De seguida, é descrita a arquitetura do Sistema de Apoio à Exploração, que faz parte desse produto. Posteriormente é analisada a plataforma de regulação ao pormenor, descrevendo as suas funcionalidades, principais algoritmos, modelo de dados que utiliza e por fim, pontos que poderão ser melhorados no sistema.

3.1 Solução chave-na-mão

A secção dos Transportes do grupo EFACEC tem afirmado uma posição marcante neste mercado, oferecendo soluções para alguns segmentos de mercado, mais concretamente os Metros Ligeiros, Ferrovias e Metros Pesados, Transportes Rodoviários e Aeroportos. No contexto deste projeto só será relevante analisar a solução apresentada para os transportes metro-ferroviários.

Neste segmento de mercado a estratégia da empresa é uma oferta de solução tipo "chave-na-mão", isto é, entregar ao cliente toda a infraestrutura necessária para colocar uma rede de metro em funcionamento [1]:

- Redes de transmissão (fibra ótica, cabo metálico e wireless);
- Sistemas de vídeo inteligente (deteção de ocorrências, intrusões, objetos abandonados, etc.);
- Informação ao público (visual e sonora);
- Plataforma VoIP de comunicações operacionais e de emergência (SOS, telefonia de exploração);
- Sistemas de localização (GPS, DGPS, RFID);
- Sistemas integrados de gestão e operação;
- Sistemas de apoio à exploração (SAE);

- Sistema de monitorização de alarmes;
- Sistema de alimentação;

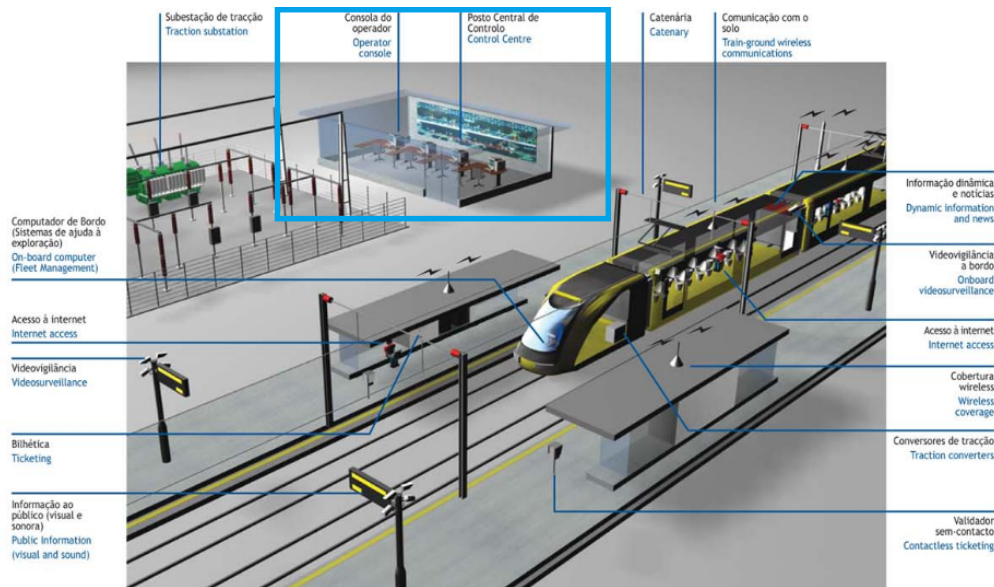


Figura 3.1: Elementos constituintes da solução chave-na-mão da EFACEC [1]

A figura 3.1 ilustra quais os sistemas que a EFACEC oferece nesta área. No quadrado azul está representado a parte que foi objeto de estudo neste trabalho, é nesse contexto que é utilizado o software de apoio à decisão, neste caso concreto não será apenas um SAD, mas sim um Sistema de Apoio à Exploração, pois engloba mais funcionalidades que vão para além de software.

3.2 Sistema de Apoio à Exploração

O Sistema de Apoio à Exploração, ou também conhecido como *Automatic Vehicle Location System*, é um conjunto de ferramentas que têm como objetivo ajudar as companhias de transporte na gestão e operação dos seus serviços.

Na EFACEC, este plataforma tem o nome de *TimeKeeper* e consiste numa aplicação central, utilizada no PCC e uma aplicação embebida em equipamentos a bordo dos veículos. As comunicações entre o dois é feita por redes rádio, mais concretamente rede TETRA, e rede WI-FI. O *TimeKeeper* do PCC recebe diversas informações acerca da localização dos veículos, ocorrência de falhas, e alarmes de segurança. A localização do veículo é dada através do sistema embebido no veículo e por pontos localizados ao longo da linha, tornando o sistema redundante de informação, e por isso mais seguro em caso de falhas.

Os operadores que se encontram no PCC recebem esta informação em tempo real, sabendo desta forma qual a localização dos veículos a todo o momento. Esta informação é apresentada em diversas interfaces gráficas, desde de sinópticos, tabelas, gráficos espaço/tempo e outros. Estes

dados permitem também dar *feedback* ao condutor do veículo, indicando-lhe se este se encontra atrasado ou adiantado. O público também é consumidor da informação vinda do *timekeeper*, pois este apresenta nos painéis presentes nas estações, o tempo estimado que falta para o próximo veículo. Na figura 3.2 esta representada a arquitetura desta plataforma.

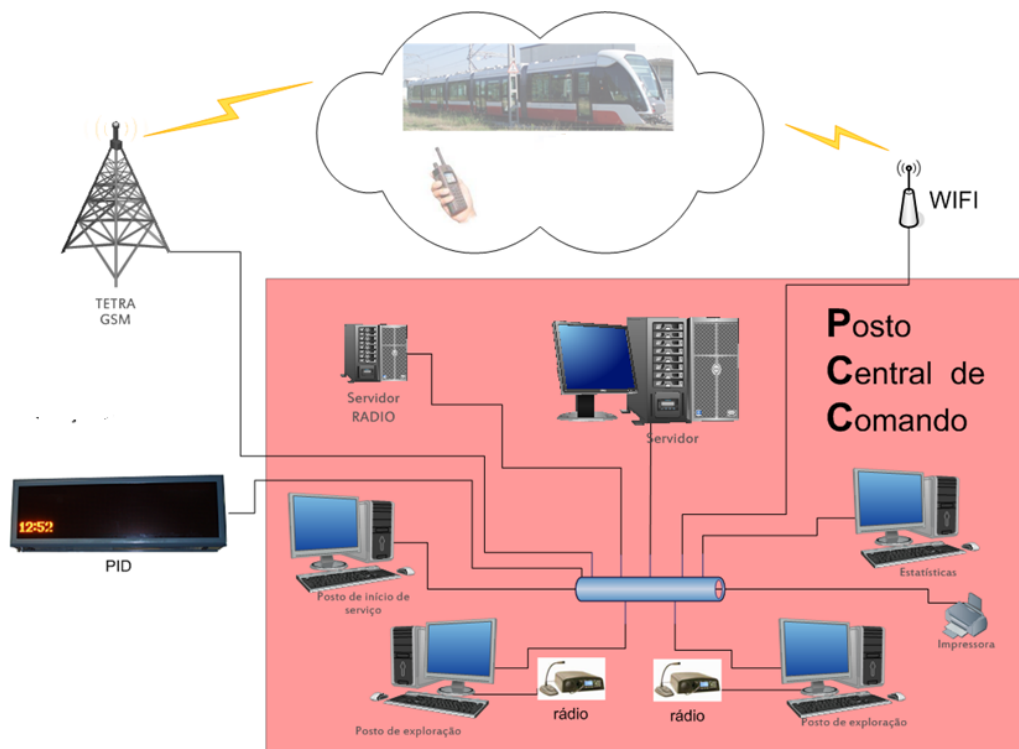


Figura 3.2: Arquitetura geral do TimeKeeper

Como é possível verificar, o sistema central desta plataforma possui diferentes interfaces ou módulos: módulo de estatística, módulo gestão de condutores, módulo de treino, módulo de regulação, sinóptico (janela principal) entre outros. No âmbito da dissertação destacaremos apenas o módulo de sinóptico e o módulo de regulação, o primeiro porque é a principal janela de monitorização e também contém informação importante para se ter uma perceção global do estado da linha, o segundo por razões óbvias merece um estudo mais aprofundado.

Em período de operação estes dois módulos são os mais utilizados no Posto Central de Comando, enquanto que os outros módulos à disposição terão mais importância na gestão e planeamento prévio. Na figura 3.3 podemos ver a janela apresentado pelo módulo de sinóptico, esta janela é alimentada com informação em tempo-real e apresenta a localização relativa dos veículos, também é possível perceber se os veículos se encontram atrasados ou adiantados através de um código de cores do símbolo que representa o veículo.

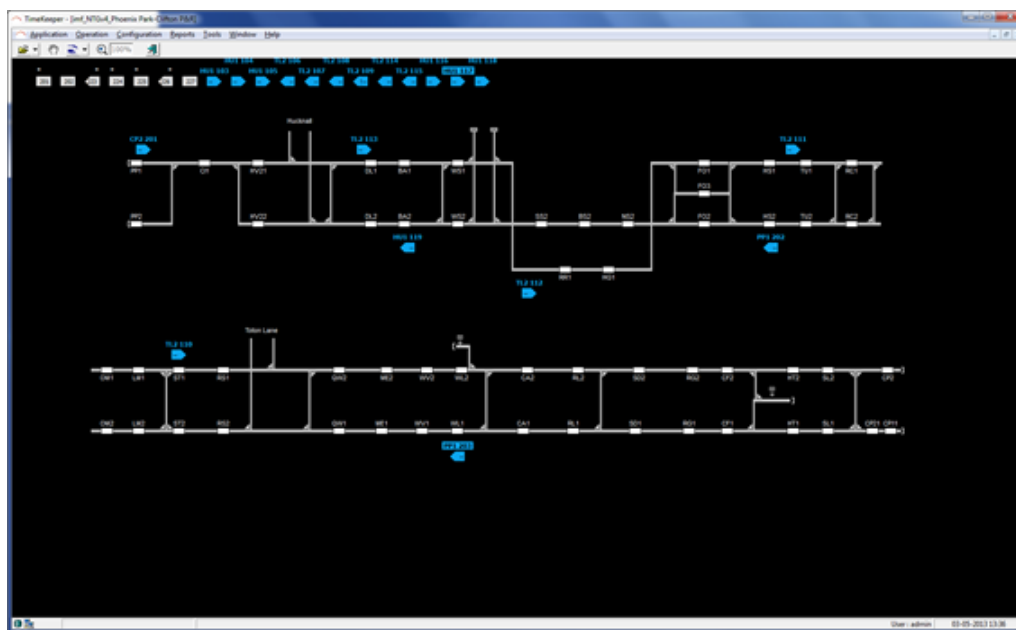


Figura 3.3: Janela principal do SAE da EFACEC

Não será relevante entrar em demasiados pormenores em relação a esta janela, pois esta tem inúmeras opções, mas no âmbito do trabalho só interessa perceber qual a informação relativa ao estado da linha e dos veículo, pois são dos fatores mais importante no processo de monitorização da rede.

3.3 Módulo de regulação

Nesta secção iremos descrever com detalhe o módulo de regulação que é considerado um dos mais importantes em todo o Sistema de Apoio a Exploração apresentado anteriormente.

O módulo de regulação é um sistema de apoio à decisão dos reguladores que se encontram no PCC tendo como principal objetivo otimizar o tráfego da sua frota pelas rotas/linhas preestabelecidas, podendo modificar os respetivos horários e itinerários para dar resposta a situações de perturbação.

Na figura 3.4, é possível observar o aspecto visual desta aplicação. Na parte central existe uma tabela, em que cada linha representa uma viagem, onde temos informação sobre a linha, o serviço, o turno, a origem e o destino da viagem, a hora de partida e chegada planeada, e o trajeto. Na lateral direita temos um conjunto de botões que representam as ações à disposição do operador. No fim desta janela temos uma tabela com informação relativa ao sucesso ou não das nossas ações.

Para desempenhar as suas funções esta ferramenta precisa de ter algumas entradas, tais como:

- o horário em uso: um horário contém a hora de chegada e tempo de paragem em cada estação, com informações sobre o número da viagem, trajeto e serviço;
- dados relativos aos trajetos permitidos pela sinalização rodoviária;

- a localização em tempo real dos veículos;
- informações relativas à rede: pontos de viragem, linhas partilhadas, tempos mínimos de percurso;
- correspondências entre linhas.

| Regulation | | | | | | | | | | | | Services | |
|----------------------------|------|---------|-------|----------------------------|--------------------------|-----------|----------|--------------------|------------------|---------|--------------|-----------|-------------------|
| Day | Line | Service | Trips | Origin | Destination | Departure | Arrival | Schedule departure | Schedule arrival | Block | Routes | Direction | Services |
| Click here to set a filter | | | | | | | | | | | | | |
| 28 | 1 | 112 | 3 | 3 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 09:45:00 | 10:45:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | New trip |
| 28 | 1 | 112 | 2 | 2 Totom Lane PK Stop 1-FL2 | Industrial-HU2 | 10:13:04 | 10:13:04 | | | FL2-HU2 | FL2-HU2-MJLP | | Remove trip |
| 28 | 1 | 118 | 2 | 2 Totom Lane PK Stop 1-FL2 | 3 Industrial-HU2 | 10:30:00 | 10:30:00 | | | FL2-HU2 | FL2-HU2-MJLP | | Change trip |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 10:38:40 | 10:38:40 | | | FL2-OP2 | FL2-OP2-MJLP | | View trip |
| 28 | 1 | 120 | 2 | 2 Totom Lane PK Stop 1-FL2 | Offhar PK Stop-OP2 | 10:50:00 | 10:50:00 | | | FL2-HU2 | FL2-HU2-MJLP | | |
| 28 | 1 | 121 | 2 | 2 Totom Lane PK Stop 1-FL2 | Offhar PK Stop-OP2 | 10:58:40 | 10:58:40 | | | FL2-HU2 | FL2-HU2-MJLP | | |
| 28 | 1 | 124 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 10:44:40 | 10:44:40 | | | FL2-OP2 | MJLP-HU2-FL2 | | Change route |
| 28 | 1 | 124 | 3 | 3 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 10:44:40 | 10:44:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 124 | 2 | 2 Totom Lane PK Stop 1-FL2 | Offhar PK Stop-OP2 | 10:44:40 | 10:44:40 | | | FL2-HU2 | MJLP-HU2-FL2 | | Shortenings |
| 28 | 1 | 114 | 3 | 3 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 10:50:00 | 10:50:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | Stop line |
| 28 | 1 | 114 | 2 | 2 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 10:58:40 | 10:58:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 3 | 3 Offhar PK Stop-OP2 | Phoenix Park-FL2 | 10:58:40 | 10:58:40 | | | OP2-HU2 | FL2-HU2-MJLP | | Terminals |
| 28 | 1 | 118 | 4 | 4 Totom Lane PK Stop 1-FL2 | Industrial-HU2 | 11:01:36 | 11:01:36 | | | FL2-HU2 | FL2-HU2-MJLP | | Out of service |
| 28 | 1 | 124 | 4 | 4 Totom Lane PK Stop 1-FL2 | Industrial-HU2 | 11:01:36 | 11:01:36 | | | FL2-HU2 | FL2-HU2-MJLP | | Block |
| 28 | 1 | 124 | 3 | 3 Industrial-HU2 | Phoenix Park-FL2 | 11:01:36 | 11:01:36 | | | OP2-FL2 | FL2-HU2-MJLP | | Recover |
| 28 | 1 | 118 | 3 | 3 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 11:19:36 | 11:21:54 | | | HU2-FL2 | MJLP-HU2-FL2 | | Siding |
| 28 | 1 | 120 | 3 | 3 Offhar PK Stop-OP2 | Phoenix Park-FL2 | 11:30:40 | 11:30:40 | | | FL2-HU2 | FL2-HU2-MJLP | | Cancel regulation |
| 28 | 1 | 120 | 2 | 2 Totom Lane PK Stop 1-FL2 | Offhar PK Stop-OP2 | 11:33:40 | 11:33:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 121 | 3 | 3 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 11:39:42 | 11:41:30 | | | HU2-FL2 | MJLP-HU2-FL2 | | Correction |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 11:44:33 | 11:44:33 | | | OP2-OP2 | MJLP-HU2-FL2 | | Alarm |
| 28 | 1 | 112 | 4 | 4 Totom Lane PK Stop 1-FL2 | Industrial-HU2 | 11:44:33 | 11:44:33 | | | FL2-HU2 | FL2-HU2-MJLP | | Cancel alarm |
| 28 | 1 | 112 | 3 | 3 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 11:52:40 | 11:54:48 | | | HU2-FL2 | MJLP-HU2-FL2 | | Headway |
| 28 | 1 | 112 | 2 | 2 Industrial-HU2 | Industrial-HU2 | 11:55:02 | 11:57:18 | | | FL2-HU2 | FL2-HU2-MJLP | | |
| 28 | 1 | 114 | 3 | 3 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 11:55:02 | 11:55:02 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 4 | 4 Totom Lane PK Stop 1-FL2 | Industrial-HU2 | 11:55:02 | 11:57:18 | | | FL2-HU2 | FL2-HU2-MJLP | | |
| 28 | 1 | 124 | 3 | 3 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 12:04:32 | 12:06:40 | | | FL2-HU2 | FL2-HU2-MJLP | | |
| 28 | 1 | 124 | 2 | 2 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 12:04:32 | 12:06:40 | | | FL2-HU2 | FL2-HU2-MJLP | | |
| 28 | 1 | 118 | 3 | 3 Industrial-HU2 | Offhar PK Stop-OP2 | 12:26:51 | 12:27:07 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 117 | 4 | 4 Totom Lane PK Stop 1-FL2 | Industrial-HU2 | 12:38:46 | 12:41:00 | | | FL2-HU2 | FL2-HU2-MJLP | | |
| 28 | 1 | 118 | 4 | 4 Totom Lane PK Stop 1-FL2 | Industrial-HU2 | 12:38:46 | 12:41:00 | | | FL2-HU2 | FL2-HU2-MJLP | | |
| 28 | 1 | 120 | 3 | 3 Offhar PK Stop-OP2 | Phoenix Park-FL2 | 12:38:46 | 12:38:46 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 2 | 2 Totom Lane PK Stop 1-FL2 | Offhar PK Stop-OP2 | 12:50:00 | 12:50:00 | | | FL2-HU2 | FL2-HU2-MJLP | | |
| 28 | 1 | 121 | 3 | 3 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 12:58:40 | 12:58:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 121 | 2 | 2 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 13:07:20 | 13:09:36 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 13:16:00 | 13:16:00 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 13:24:40 | 13:24:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 13:33:20 | 13:33:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 13:42:00 | 13:42:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 13:50:40 | 13:50:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 14:00:00 | 14:00:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 14:08:40 | 14:08:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 5 | 5 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 14:17:20 | 14:17:20 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 14:26:00 | 14:26:00 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 14:34:40 | 14:34:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 14:43:20 | 14:43:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 14:52:00 | 14:52:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 15:00:40 | 15:00:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 15:09:20 | 15:09:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 15:18:00 | 15:18:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 5 | 5 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 15:26:40 | 15:26:40 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 15:35:20 | 15:35:20 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 15:44:00 | 15:44:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 15:52:40 | 15:52:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 16:01:20 | 16:01:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 16:10:00 | 16:10:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 16:18:40 | 16:18:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 16:27:20 | 16:27:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 5 | 5 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 16:36:00 | 16:36:00 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 16:44:40 | 16:44:40 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 16:53:20 | 16:53:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 17:02:00 | 17:02:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 17:10:40 | 17:10:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 17:19:20 | 17:19:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 17:28:00 | 17:28:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 17:36:40 | 17:36:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 5 | 5 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 17:45:20 | 17:45:20 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 17:54:00 | 17:54:00 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 18:02:40 | 18:02:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 18:11:20 | 18:11:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 18:20:00 | 18:20:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 18:28:40 | 18:28:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 18:37:20 | 18:37:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 18:46:00 | 18:46:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 5 | 5 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 18:54:40 | 18:54:40 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 19:03:20 | 19:03:20 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 19:12:00 | 19:12:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 19:20:40 | 19:20:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 19:29:20 | 19:29:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 19:38:00 | 19:38:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 19:46:40 | 19:46:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 19:55:20 | 19:55:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 5 | 5 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 20:04:00 | 20:04:00 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 20:12:40 | 20:12:40 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 20:21:20 | 20:21:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 20:30:00 | 20:30:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 20:38:40 | 20:38:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 20:47:20 | 20:47:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 20:56:00 | 20:56:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 21:04:40 | 21:04:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 5 | 5 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 21:13:20 | 21:13:20 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 21:22:00 | 21:22:00 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 21:30:40 | 21:30:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 21:39:20 | 21:39:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 21:48:00 | 21:48:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 114 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 21:56:40 | 21:56:40 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 5 | 5 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 22:05:20 | 22:05:20 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 118 | 4 | 4 Industrial-HU2 | Totom Lane PK Stop 1-FL2 | 22:14:00 | 22:14:00 | | | HU2-FL2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 5 | 5 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 22:22:40 | 22:22:40 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 120 | 4 | 4 Phoenix Park-FL2 | Offhar PK Stop-OP2 | 22:31:20 | 22:31:20 | | | OP2-OP2 | MJLP-HU2-FL2 | | |
| 28 | 1 | 112 | 5 | | | | | | | | | | |

Figura 3.4: Janela principal do módulo de regulação da EFACEC

As entradas do sistema são usados para calcular um novo horário tendo em conta uma ação a realizar pelo controlador, dentro de um conjunto de ações disponibilizadas na barra lateral direita da figura 3.4. O horário de saída será um horário que deverá cumprir todas as restrições que existem num sistema deste tipo (diferença de tempo entre veículos, tempos mínimos de paragem, tempos mínimos de viragem, etc). Por fim, os veículos serão automaticamente informados acerca das mudanças feitas nos seus serviços e as respetivas viagens afetadas.

Este sistema não é um sistema automático e tem como função apenas auxiliar o regulador nas suas decisões. Por esse motivo, o sistema permite ao utilizador proceder a qualquer ação, mesmo que esta possa parecer menos lógica. Por isso a boa utilização deste sistema, e também a qualidade dos seus resultados depende em grande parte da experiência do operador. Este deverá ter um bom conhecimento da linha, pontos importantes da mesma, tempos de viagem nas várias secções como também as regras básicas da operação. No entanto, o sistema lança alguns avisos quando deteta ações que não cumpram os requisitos, mas, são à mesma enviadas para os veículos em circulação.

3.3.1 Manobras de regulação

Este módulo é normalmente utilizado quando há uma secção da linha que está bloqueada por algum motivo, quando existe um atraso significativo no horário planeado, ou quando é necessário colocar mais ou menos veículos devido a um evento não esperado. Para resolver estes problemas existem um conjunto de ações à disposição do operador, ou como normalmente são conhecidas manobras de regulação. Serão apresentadas todas as manobras existentes no *software* da EFACEC de uma forma breve, pois não seria possível analisar todas estas ações de uma forma muito extensiva por limitações de tempo. Então, as manobras ao dispor do operador são:

- **Serviços** – esta função serve para o utilizador criar serviços, tirar serviços, fazer mudanças de viagens entre 2 serviços;
- **Nova Viagem** – esta função serve para adicionar novas viagens a serviços já existentes, podendo ser criadas uma ou mais viagens em simultâneo;
- **Retirar Viagem** – esta função serve para retirar uma ou mais viagens seleccionadas;
- **Encurtamentos** – esta função serve para definir um começo numa plataforma mais adiante do que o inicial ou para definir um fim numa plataforma antes do que estava definido como final;
- **Alterar Viagem** – esta função permite modificar os dados relativos relacionados com saídas, tempos de paragem e tempos de percurso de uma viagem;
- **Tempo de Paragem** – esta função serve para o utilizador poder alterar o tempo de paragem numa ou mais plataformas, afetando uma ou mais viagens;
- **Terminais** – esta função serve para incrementar ou decrementar tempo às saídas num determinado período, ou seja alterar o horário de partida;
- **Fora de Serviço** – esta função serve para colocar uma determinada plataforma fora de serviço;
- **Retenção** – esta função serve para ordenar que um determinado veículo fique bloqueado numa determinada plataforma;
- **Alterar Trajeto** – esta função permite alterar todo o trajeto de uma ou várias viagens por outro trajeto;
- **Anular regulações** – esta função serve para anular alguma das ações antes tomadas;
- **Recuperação** – esta função tem como objetivo, numa janela de tempo seleccionada, estabilizar a diferença de tempo entre os veículos;

- **Deslizamento** – esta função é utilizada pelo regulador para recuperar o tempo de atraso, selecionando uma viagem que esteja atrasada; para tal, o sistema vai deslizar o horário planeado da viagem para o horário real, deixando este veículo de se apresentar atrasado, em seguida o sistema poderá acrescentar esse tempo a todas as viagens subsequentes ou então tentar recuperar esse tempo adicionado;
- **Conexão** – funcionalidade que consiste em verificar se as viagens de cada serviço de transporte estão unidas, identificando-as e criando novas viagens de conexão entre terminais entre essas viagens não unidas, caso o utilizador o pretenda;
- **Regulação por intervalo** – construção de um horário que espase igualmente no tempo os serviços escolhidos pelo regulador;
- **Intervalo de carrossel** – esta função serve para o regulador poder criar viagens entre duas plataformas espaçadas igualmente no tempo durante um determinado período; poderá ser utilizada em casos onde haja avarias em partes da linha.

3.3.2 Algoritmos de otimização

As manobras acima enumeradas são normalmente utilizadas com o intuito de reduzir o impacto de uma perturbação: falha de energia em parte da via, atraso, afluência de mais passageiros, avaria no veículo, entre outras.

Algumas destas manobras invocam um algoritmo, que tem como função procurar construir um horário o mais semelhante possível com o horário planeado.

De uma forma simplificada o algoritmo procede de uma forma iterativa, e em cada interação tenta reduzir o tempo de viragem em terminus, o tempo de paragem nas estações seguintes ao atraso, fazendo o mesmo na viagem seguinte na interação seguinte. O algoritmo procede desta forma até alcançar uma solução, ou até o número de interações máximas ser alcançado. Uma solução neste contexto será um horário sem conflitos entre viagens e com o atraso inicial diluído por um conjunto de viagens.

Devido às limitações computacionais, por vezes o limite de interações impede que o algoritmo consiga encontrar uma solução. Nestas situações o módulo avisa que existem conflitos provocados pela manobra utilizada e que estes deverão ser resolvidos pelo operador.

3.3.3 Limitações da plataforma

A plataforma descrita foi desenvolvida por uma equipa de experientes engenheiros, e está em funcionamento em vários metros espalhados pelo mundo. Os resultados apresentados por esta aplicação são considerados bons e por estes motivos é complicado estar-se a falar de limitações, e mais complicado ainda é identificar essas limitações. O que irei abordar de seguida são apenas alguns itens que parecem poder integrar a plataforma tornando-a ainda melhor.

O primeiro ponto a melhorar passa por reestruturar o interface gráfico da aplicação. Quando olhamos para a aplicação a janela apresentada tem um aspecto um pouco desatualizado, figura 3.4.

Também se pode referir que na informação tabular esta poderia estar apresentada de uma forma diferente, colocando em cada linha o horário da mesma forma que são apresentados os horários ao público. Por baixo, de cada célula poderia ser colocado o atraso que existe em relação ao horário planeado. Ainda em relação ao interface, penso que ter que se abrir uma janela específica para cada ação a realizar, poderá ser substituída em certos casos por um *slider*.

O segundo ponto, e talvez o mais importante é o facto do sistema não permitir simular cenários em tempo real, ou seja, todas as ações realizadas são enviadas para a base de dados e para os veículos em circulação. Este facto torna impossível para o operador poder comparar diferentes ações que poderia tomar quando quiser dar resposta a um problema na rede. É também um facto que a comparação de diferentes cenários é algo complicado. No entanto, como foi dito na secção 2.1.2, a forma mais utilizada neste meio passa por recorrer aos diagramas espaço/tempo. Por isso penso que seria importante ter capacidade de simular em tempo real, e receber o *feedback* no gráfico a solução, devendo o sistema ser capaz de gerar múltiplos cenários em múltiplos diagramas espaço/tempo.

O terceiro ponto é algo complementar ao ponto anterior. Além da possibilidade de simular, em algumas manobras de regulação, tal como deslizamento, deveria ser possível para o operador de alguma forma passar ao sistema como quer que essa ação seja computorizada.

O quarto ponto tem a ver com o facto de por vezes o sistema não dar soluções, aqui o sistema poderia ao invés de deitar fora todo o trabalho computorizado, este poderia dar a solução parcial.

O quinto ponto identificado não tem diretamente a ver com o módulo de regulação mas sim com a forma como foi desenvolvido o mesmo. Tendo em atenção o número de pessoas envolvidas na construção do mesmo, e a complexidade do sistema, penso que se devia adotar um padrão de desenvolvimento, tal como Model View ViewModel, ou outro.

Capítulo 4

Caso de estudo

Depois de analisado a solução existente na empresa, foi necessário definir os limites do problema a tratar, para ser possível sugerir uma alternativa exequível. Neste capítulo é descrito esse problema de uma forma objetiva. Para tal recorre-se à linguagem matemática com o objetivo de ser possível representar toda a dinâmica do problema. Este capítulo permitirá ao leitor perceber o procedimento desenvolvido para recuperar um atraso.

4.1 Problema a tratar

Neste ponto já somos capazes de perceber a complexidade do sistema de regulação que o grupo EFACEC possui. E isto, aliado ao tempo disponível fez com que fosse necessário definir fronteiras e algumas simplificações do problema. É importante referir que não é objetivo deste projeto de dissertação refazer na totalidade o software que foi alvo de estudo, nem tal seria possível.

Como se sabe uma rede de metro poderá ter diversas configurações, diversos números de linhas, vias, estações, cruzamentos, entre outras características. Por estes motivos é necessário definir qual será o tipo de configuração a considerar.

Neste projeto, a título ilustrativo, considerou-se uma linha de metro com 2 vias, 7 estações, sendo que a circulação de veículos é feita em sentidos opostos em cada uma destas. Este tipo de configuração é das mais comuns em todo o mundo, pois é a base de qualquer rede de metropolitano, não existirão pontos de cruzamento de veículos, ou seja cada sentido terá a sua via fixa, e só nas estações de fim de via se poderá dar a volta. Esta configuração é semelhante à Linha Amarela do Metro do Porto, mas com menos estações.

Em seguida selecionamos uma manobra de regulação para estudo e implementação, a escolha recaiu sobre a manobra de deslizamento, esta foi escolhida pois como foi dito em cima, esta é utilizada com a finalidade de reduzir o impacto de um eventual atraso no horário planeado sem ter de recorrer à colocação de mais veículos na linha, ou criação de novas viagens. Esta é também uma das ações mais utilizadas na operação do dia-a-dia, pois é a forma mais fácil que o operador tem de recuperar um atraso sem ter que se esforçar em arquitetar outra solução. Poderiam ter sido

escolhidas outras manobras, mas muitas delas só são usadas em situações muito específicas e que pouco interesse tem estudar.

Na secção 3.3.3, foram enumeradas possíveis melhorias que poderiam ser feitas no sistema da EFACEC, e foi com essas sugestões no horizonte que se desenvolveu a solução. Um objetivo deste trabalho passa por saber se realmente as melhorias sugeridas serão ou não implementáveis e se terão resultados satisfatórios.

4.2 Notação

Existem muitas formas para abordar este tipo de problema, umas mais sofisticadas do que outras. No entanto, para os objetivos do trabalho optou-se por um algo simples tentando cumprir os pressupostos assumidos na secção 4.1, deste documento.

Então, neste caso estaremos a falar de um linha de metro com duas vias, sendo utilizada cada uma delas para sentidos opostos. Os índices utilizados para definir as estações serão os mesmos em ambos os sentidos. Para ser possível distinguir qual o sentido ao qual nos estamos a referir existe um índice auxiliar. É importante não confundir esta nomenclatura com um problema semelhante que em vez de ter uma via independente para cada sentido, tem apenas uma. Na realidade não haverá grandes diferenças, tendo apenas mais algumas restrições a esse nível.

As variáveis do modelo estão divididas em variáveis relativas ao *planeado*, e variáveis relativas ao *operacional*, sendo o planeado o horário que previamente foi estipulada para a atividade e o horário operacional, o horário que está realmente a ser cumprido pelos veículos ou que foi modificado em plena atividade.

Planeado

HCP_{ij}^1 - Hora de chegada planeado do veículo “i” à estação “j” no sentido 1;

HCP_{ij}^2 - Hora de chegada planeado do veículo “i” à estação “j” no sentido 2;

HPP_{ij}^1 - Hora de partida planeado do veículo “i” da estação “j” no sentido 1;

HPP_{ij}^2 - Hora de partida planeado do veículo “i” da estação “j” no sentido 2;

TPP_{ij}^1 - Tempo de paragem planeado do veículo “i” na estação “j” no sentido 1;

TPP_{ij}^2 - Tempo de paragem planeado do veículo “i” na estação “j” no sentido 2;

$TVP_{i(j \rightarrow j+1)}^1$ - Tempo de viagem do veículo “i” da estação j a j+1 no sentido 1;

$TVP_{i(j+1 \rightarrow j)}^2$ - Tempo de viagem do veículo “i” da estação j+1 a j no sentido 2;

TPV_{ij}^1 - Tempo de dar a volta do veículo “i” na estação “j” no sentido 1;

TPV_{ij}^2 - Tempo de dar a volta do veículo “i” na estação “j” no sentido 2;

Operacional

HCO_{ij}^1 - Hora de chegada operacional do veículo “i” à estação “j” no sentido 1;

HCO_{ij}^2 - Hora de chegada operacional do veículo “i” à estação “j” no sentido 2;

HPO_{ij}^1 - Hora de partida operacional do veículo “i” da estação “j” no sentido 1;

HPO_{ij}^2 - Hora de partida operacional do veículo “i” da estação “j” no sentido 2;

TPO_{ij}^1 - Tempo de paragem operacional do veículo “i” na estação “j” no sentido 1;
 TPO_{ij}^2 - Tempo de paragem operacional do veículo “i” na estação “j” no sentido 2;
 $TVO_{i(j \rightarrow j+1)}^1$ - Tempo de viagem operacional do veículo “i” entre a estação “j” e “j+1” no sentido 1;
 $TVO_{i(j+1 \rightarrow j)}^2$ - Tempo de viagem operacional do veículo “i” entre a estação “j+1” e “j” no sentido 2;
 TOV_{ij}^1 - Tempo de dar a volta do veículo “i” na estação “j” no sentido 1;
 TOV_{ij}^2 - Tempo de dar a volta do veículo “i” na estação “j” no sentido 2;

4.3 Restrições

Na secção anterior, foram mostradas todas as variáveis identificadas e que terão relevância para a abordagem ao tema. Em seguida, mostraremos as restrições existentes neste tipo de sistemas. Iremos apenas representar as equações referentes aos horários operacionais, no entanto estas restrições aplicam-se também ao horário planeado.

1. Tempos mínimos de paragem na estação.

Esta restrição resulta de parâmetros especificados pela linha de metro, e tem por base um tempo mínimo para os passageiros saírem ou entrarem no veículo.

$$TPO_{ij}^1 > TPOmin_{ij}^1 \text{ (Tempo mínimo de paragem na estação "j" no sentido 1)}$$

$$TPO_{ij}^2 > TPOmin_{ij}^2 \text{ (Tempo mínimo de paragem na estação "j" no sentido 2)}$$

2. Tempo mínimo de percurso.

Devido às restrições ferroviárias, como também, ao limite de velocidade máxima que um veículo consegue alcançar, existe um tempo mínimo para ser possível percorrer duas estações adjacentes.

$TVO_{i(j \rightarrow j+1)}^1 > TVOmin_{i(j \rightarrow j+1)}^1$ (O veículo tem um tempo mínimo para percorrer um percurso entre a estação “j” e “j+1”, que é o correspondente a andar à velocidade máxima no sentido 1 entre essas duas estações.)

$TVO_{i(j+1 \rightarrow j)}^2 > TVOmin_{i(j+1 \rightarrow j)}^2$ (O veículo tem um tempo mínimo para percorrer um percurso entre a estação “j+1” e “j”, que é o correspondente a andar à velocidade máxima no sentido 2 entre essas duas estações.)

3. Diferença de tempo mínima entre veículos.

Os veículos adjacentes devem circular com uma diferença de tempo mínima entre eles. Esta restrição resulta também de medidas de segurança, e também tende a ser um parâmetro da própria linha. Esta restrição garante que não exista mais do que um veículo ao mesmo tempo na mesma estação.

$HPO_{(i+1)j}^1 - HCO_{ij}^1 > THmin$ (É necessário garantir que os veículos circulam com uma diferença de tempo mínima, esta restrição também garante que não existem dois veículos no mesmo local, ao mesmo tempo. NOTA: THmin será o tempo de headway.)

4. Tempo mínimo para dar a volta nas estações de terminal.

Quando um veículo acaba uma viagem num sentido, existe um tempo mínimo para este poder começar uma viagem no sentido contrário. Este tempo tem que ser suficiente para o veículo passar para a outra via e para o condutor chegar à cabine correspondente.

$HOP_{iN}^2 - HCO_{iN}^1 > TOVmin_{iN}^1$ (Garante que veículo só parta da estação N, sendo N a última estação no sentido 1, e a primeira no no sentido 2, depois deste chegar à estação N no sentido 1 mais o tempo mínimo que será necessário para o veículo dar a volta.)

4.4 Recuperação de atrasos

Numa linha de metro, com duas vias, um horário só será valido se cumprir as restrições anteriores, e se tal não acontecer, estaremos perante um horário com conflitos.

Estas formulações tiveram como objetivo perceber de que forma os horários podem ser construídos e quais são as variáveis de que poderão depender. Em seguida, são mostradas algumas observações a fazer, que devem deixar mais a clara toda a dinâmica envolvida neste problema.

A primeira observação a ser feita, é algo óbvio, e que diz que o horário de partida de um veículo da estação j, é igual ao horário de chegada de mesmo veículo à estação j mais o tempo de paragem em j.

$$HPO_{ij}^1 = HCO_{ij}^1 + TPO_{ij}^1$$

$$HPO_{ij}^2 = HCO_{ij}^2 + TPO_{ij}^2$$

A segunda observação que se pode fazer é que, dentro da mesma viagem, o horário de chegada à estação j, poderá ser definido a partir do horário de chegada à estação j-1, mais o tempo de paragem na estação j-1, mais o tempo de viagem entre a estação j-1 e j.

$$HCO_{ij}^1 = HCO_{ij-1}^1 + TPO_{ij-1}^1 + TVO_{i(j-1 \rightarrow j)}^1$$

$$HCO_{ij}^2 = HCO_{ij+1}^2 + TPO_{ij+1}^2 + TVO_{i(j+1 \rightarrow j)}^2$$

A terceira observação a ser feita terá como pano de fundo a ligação de 2 viagens dentro do mesmo serviço, relembro que serviço será um conjunto de viagens executadas de forma sequencial, ou seja, a viagem número n+1 só poderá começar depois da viagem n terminar. Então o

horário de partida da viagem $n+1$ será igual ao horário de chegada à última estação da viagem n , mais o tempo de dar a volta.

$$HPO_{iN}^2 = HCO_{iN}^1 + TVO_{iN}^1$$

$$HPO_{iN}^1 = HCO_{iN}^2 + TVO_{iN}^2$$

A quarta observação esta relacionada com veículos de serviços distintos que seguem uns atrás dos outros, então o horário de chegada à estação n de um serviço X , poderá ser dado pelo o horário de chegada à estação n do serviço Y , que segue à frente do serviço X , mais tempo que separa os dois veículos.

$$HCO_{ij}^1 = HCO_{(i+1)j}^1 + TH_{i \rightarrow i+1}^1$$

$$HCO_{ij}^2 = HCO_{(i+1)j}^2 + TH_{i \rightarrow i+1}^2$$

Estes quatro pontos podem parecer algo óbvio e simples, no entanto captam toda a dinâmica de um horário e mais importante fazem nos perceber quais as dependências dos horários entre si. A partir destas formulações podemos perceber quais são as formas de recuperar um atraso num horário, e não são muitas. Se for analisada com atenção as equações acima percebemos se acontecer um atraso, só poderemos recuperar este atraso se:

- diminuirmos os Tempos de Paragem;
- diminuirmos os Tempos de Viagem;
- diminuirmos os Tempos de Viragem.

No entanto, ao procurarmos diminuir estes tempos sem impor limites, iremos sem sombra de dúvida quebrar alguma das restrições antes definidas. E por isso, decidiu-se definir limites para essas diminuições, e a esse resultado demos o nome de folgas do horário. Então, de forma enumerativa apresentamos as folgas de um horário:

- Tempo de Paragem Planeado - Tempo mínimo de Paragem = Folga de Tempo de Paragem

$$TPP_{ij}^1 - TPPmin_{ij}^1 = FTP_{ij}^1$$

$$TPP_{ij}^2 - TPPmin_{ij}^2 = FTP_{ij}^2$$

- Tempo de Viagem Planeado - Tempo mínimo de Viagem = Folga de Tempo de Viagem

$$TPP_{ij \rightarrow j+1}^1 - TPPmin_{ij \rightarrow j+1}^1 = FTViag_{ij \rightarrow j+1}^1$$

$$TVP_{ij+1 \rightarrow j}^2 - TPPmin_{ij+1 \rightarrow j}^2 = FTViag_{ij+1 \rightarrow j}^2$$

- Tempo de Viragem Planeado - Tempo mínimo de Viagem = Folga de Tempo de Viragem

$$TPV_{iN}^1 - TPPmin_{iN}^1 = FTVir_{iN}^1$$

$$TPV_{iN}^2 - TPPmin_{iN}^2 = FTVir_{iN}^2$$

Com base nestes conceitos é possível tecer alguns comentários sobre um horário. Quando o valor destas folgas é muito reduzido, é porque estamos perante horários muito rígidos e que não terão grande possibilidade para recuperar de um atraso, se pelo contrário o valor destas folgas for considerável então está-se perante um horário mais flexível e com maior capacidade de resposta a eventos adversos. É importante referir que em maioria das companhias de redes de metropolitano, os horários são construído tendo em atenção este facto.

Capítulo 5

Método de resolução

Neste capítulo é apresentado o método de resolução do problema. Inicialmente é descrito os algoritmos que foram propostos para implementar a manobra de deslizamento e de forma mais generalizada, algoritmos que tem como objetivo recuperar um atraso. Para tal, antes foram analisados os atrasos do ponto de vista temporal, e foi feita uma tentativa de perceber o impacto do mesmos no resto do horário. De seguida são descritos dois algoritmos e quais os objetivos dos mesmos.

5.1 Introdução

No capítulo anterior apresentou-se o modelo matemático do problema e as dinâmicas do problema: restrições e a forma de recuperar de um atraso. Como dito antes, uma linha de metro está sujeita a inúmeros fatores externos, como por exemplo acidentes, avarias, maior afluência de passageiros, entre outros. Todos eles poderão criar problemas no planeado, e uma das consequências mais comuns são os atrasos, algo que não é do agrado dos utilizadores deste tipo de transportes.

Um sistema de regulação tem como um dos objetivos ajudar o operador a conseguir diminuir os efeitos desses eventos nos horários. Como foi apresentado no capítulo 3, este tipo de SAD tem normalmente inúmeras ações ao dispor do operador. No entanto, a manobra de deslizamento é sem dúvida a maneira mais fácil, mais utilizada e menos exigente do ponto de vista do operador e da própria companhia de transportes de executar. E isto, porque, o princípio desta ação é adicionar a todas a viagens posteriores à viagem atrasada, o valor do atraso. É verdade que assim todos os veículos ficarão atrasados em relação ao horário planeado, mas é a forma mais fácil de repor um horário exequível. É relevante referir que quando acontece um atraso numa determinada viagem, esta poderá impedir que todo o horário planeado deixe de ser possível de cumprir, e se tal acontece, não será possível para o operador voltar a construir o novo horário de raiz.

Mas como já foi dito antes, os horários planeados são construídos tendo em conta que imprevistos podem acontecer, por isso em maioria das implementações da manobra de deslizamento, inclusivamente a implementação no sistema de regulação da EFACEC, num segundo momento o sistema tenta diluir os tempos adicionados nas viagens seguintes, diminuindo os tempos de folga,

referidos na secção 4.4. Então, se o valor do atraso não for muito maior do que esses tempos de folga, ao fim de algumas viagens, poderemos ter um horário operacional muito parecido, ou até igual ao planeado.

É fácil perceber que podem existir várias formas de implementar esta manobra, o objetivo deste projeto é implementar uma forma de o fazer, tendo em mente que queremos sempre que o nosso sistema nos entregue uma solução e que seja possível ter um algoritmo com capacidade de gerar diferentes soluções de acordo com a necessidade do operador. No caso concreto da EFACEC, o sistema implementa esta manobra da seguinte forma:

- desliza todas as viagens no valor do atraso;
- se tiver a opção de otimização ativa, vai tentar diluir o atraso nas próximas viagens, gastando de forma cega os tempos de folga que o horário tem, até este tempo de atraso ser zero;

Numa primeira análise poderá parecer a forma normal de proceder, contudo com esta abordagem o horário que resultará deste algoritmo será um horário rígido sem tempos de folgas, e portanto sem forma de recuperar novos atrasos que possam acontecer nas viagens seguintes. A outra crítica que se pode fazer é o facto de o atraso ser diluído sempre da mesma forma, não havendo forma de interação com o operador.

5.2 Classificação de atrasos

Numa primeira fase tentou-se perceber qual será o impacto de um atraso de um veículo no resto da rede e no horário que estava planeado. Se se olhar para o comportamento dinâmico da rede, um atraso num veículo poderá afetar ou não toda a rede. Analisando vários diagramas de viagens, chega-se à conclusão que um atraso que não ultrapasse a diferença de tempo do veículo que segue atrás poderá ser em parte recuperado apenas nas viagens seguintes do mesmo serviço, pois este não criará nenhum conflito com o horário planeado dos outros serviços. É necessário referir que quando nos referimos a ultrapassar a diferença de tempo do veículo que segue atrás, será a diferença de tempo em relação ao seu horário planeado, pois como já foi dito, é tecnicamente impossível haver ultrapassagens de veículos na linha.

Ao avaliarmos o atraso previamente a executarmos qualquer tipo de ajuste no horário, poderemos desta forma manter uma maior estabilidade no horário planeado. Desta forma, iremos manter os horários definidos dos outros serviços, com a exceção do serviço onde ocorreu o atraso. É também importante referir que com esta classificação, poderemos poupar algum poder computacional, pois será apenas alterada um serviço ao invés de todo o horário. Então decidiu-se classificar os atrasos em 2 tipos:

Tipo 1

Não ultrapassa a diferença de tempo que existe entre o horário real do veículo atrasado e o horário planeado do que segue atrás. E neste caso a recuperação do mesmo irá ser feita utilizando o algoritmo 1. Na imagem 5.1, podemos ver 2 serviços planeados representados pelas linha azul

e vermelha, a tracejado temos o horário operacional que o serviço vermelho realizou, a seta verde representa o atraso do mesmo à chegada do terminal, e a seta preta representa a diferença planeada entre os dois serviços.

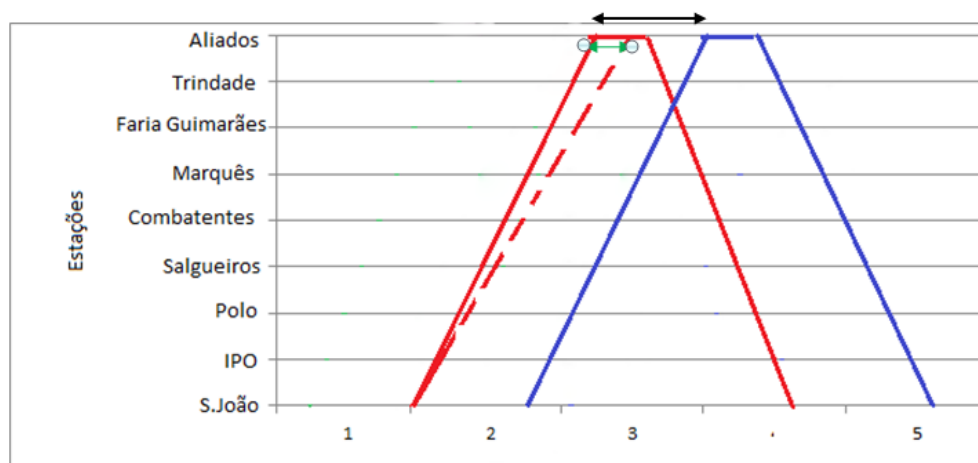


Figura 5.1: Representação gráfica de atraso Tipo 1

Tipo 2

O valor do atraso é maior que a diferença de tempo que existe entre o veículo atrasado e o horário planeado do veículo que segue atrás. Este tipo de atraso irá afetar mais serviços e por isso é preciso tratar a sua recuperação de forma global. Na imagem 5.2, mostra-se o exemplo deste tipo de atraso, volto a destacar que a diferença de tempo será em relação ao horário planeado e não ao horário real, pois se fosse em relação ao real, estaríamos perante um conflito de ultrapassagem. Seguindo a mesma lógica da figura anterior, temos 2 serviços planeados, representados pelas linhas vermelha e azul, a vermelho tracejado temos o horário operacional realizado pelo veículo do serviço vermelho, como podemos ver a seta verde é maior do que a seta preta.

5.3 Algoritmos

Ambos os algoritmos terão como entradas os horários planeados e o atraso a recuperar. Esta recuperação será pedida pelo utilizador. Em ambos os algoritmos ainda há um parâmetro de entrada dado pelo utilizador que tem como objetivo captar de que forma o utilizador pretende que o atraso seja diluído nas próximas viagens.

5.3.1 Coeficiente de recuperação

Este parâmetro será introduzido pelo utilizador e deverá refletir a forma como o utilizador quer recuperar o atraso, ou seja, de uma forma rápida ou de uma forma mais gradual.

Como foi explicado no capítulo anterior, existem tempos de folga nos horários, e é a partir da redução desses tempos que poderemos recuperar os atrasos. No entanto esses tempos poderão ser

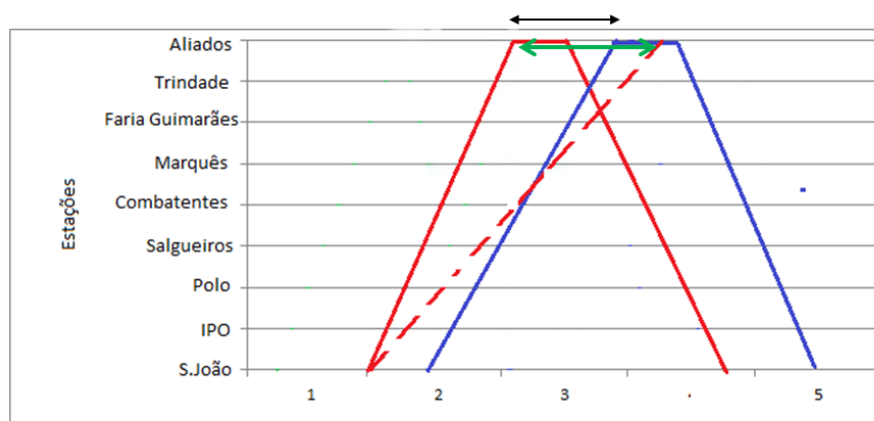


Figura 5.2: Representação gráfica de atraso Tipo 2

gastas de infinitas formas, poderá ser interessante diluir o atraso em poucas viagens, ou então diluir o atraso em muitas viagens. Ou até testar vários cenários e ver o que mais se adapta à vontade do operador.

Entenda-se que uma recuperação rápida, será tentar fazer com que o horário real volte a estar o mais rapidamente possível semelhante com o horário planeado, e lento o horário real vai tentando abater o atraso ao longo do tempo de uma forma gradual.

Existem múltiplas formas para implementar este comportamento, a forma escolhida passou por algo relativamente simples, pois o objetivo é tentar reduzir o trabalho computacional, pois como dito inicialmente, existem restrições de tempo apertadas para tomada de decisão. Então, o valor de entrada irá ser um valor a variar entre 0 e 0,5. Esse valor irá ser subtraído ao valor 1, e esse resultada será o nosso coeficiente de recuperação inicial. Este coeficiente irá multiplicar pelas folgas de tempo que cada viagem tem, variando o valor a recuperar dependendo do coeficiente inicial. No entanto desta forma a recuperação iria ser em valor percentual sempre igual para todas as viagens, e não foi esse o comportamento que nos propusemos a implementar. E para corrigir essa falha, este valor irá ser dividido por um valor N em cada iteração do algoritmo. O valor de N irá começar em 1 e por cada iteração ser-lhe-á somado 0,5.

| Número de iterações | Coefficiente de recuperação | Coefficiente de recuperação corrigido | Tempo disponível (seg) | Tempo utilizado(seg) |
|---------------------|-----------------------------|---------------------------------------|------------------------|----------------------|
| 1 | 0,6 | 0,6 | 100 | 60 |
| 2 | 0,4 | 0,4 | 100 | 40 |
| 3 | 0,3 | 0,3 | 100 | 30 |
| 4 | 0,24 | 0,24 | 100 | 24 |
| 5 | 0,2 | 0,2 | 100 | 20 |
| 6 | 0,171428571 | 0,2 | 100 | 20 |
| 7 | 0,15 | 0,2 | 100 | 20 |
| 8 | 0,133333333 | 0,2 | 100 | 20 |
| 9 | 0,12 | 0,2 | 100 | 20 |
| 10 | 0,109090909 | 0,2 | 100 | 20 |

Tabela 5.1: Simulação da variação do coeficiente de recuperação em função do número de iterações, com um valor de entrada inicial de 0.4

Na tabela 5.1, apresenta-se um exemplo da variação deste coeficiente em cada interação. Mas como é possível ver, a tabela possui uma coluna com um coeficiente de velocidade de recuperação corrigido, a razão para tal prende-se com a dinâmica do problema, pois não fará sentido tentar recuperar um atraso, utilizando valores extremamente baixos, por isso foi decidido optar pela existência de um valor mínimo, neste caso será 0.2, ou seja 20% do tempo que poderemos subtrair a subtrair. A tabela também mostra qual é o impacto deste coeficiente nas folgas a utilizar em cada interação.

5.3.2 Algoritmo 1

Este primeiro algoritmo será invocado quando a avaliação do atraso o classificar como um atraso Tipo 1. Por isso, este algoritmo só precisa de saber qual é a viagem de entrada, o respetivo atraso e quais as seguintes viagens desse serviço. Além destas entrada ainda necessitamos de saber de que forma o operador quer ver o atraso diluído, que será traduzida pelo coeficiente de recuperação.

Então, em forma procedimental, o algoritmo vai:

0. Deslizar todas as viagens do serviço atrasado no valor do atraso.
1. Guardar o valor do atraso.
2. Calcular as folgas do serviço em questão.
3. Na primeira viagem do serviço irá diminuir o tempo de paragem nas estações, tempo de viagem e tempos de viragem para a próxima viagem. O valor da diminuição será o valor das folgas calculadas anteriormente a multiplicar pelo coeficiente de recuperação correspondente ao número de interação em que o algoritmo se encontra. Se o valor a diminuir for superior ao tempo de atraso, então só será diminuído no valor do atraso.
4. Em seguida verificamos qual é o valor do novo atraso, se houver voltamos ao 2.

5. É repetido este conjunto de passos até o valor de Atraso ser 0. Ou então não haver mais viagens.

6. Por fim é mostrada a solução no diagrama de viagens.

Os passos anteriores têm a lógica base do algoritmo. De forma complementar, na figura 5.3 podemos ver o fluxograma do algoritmo, descrito com maior detalhe.

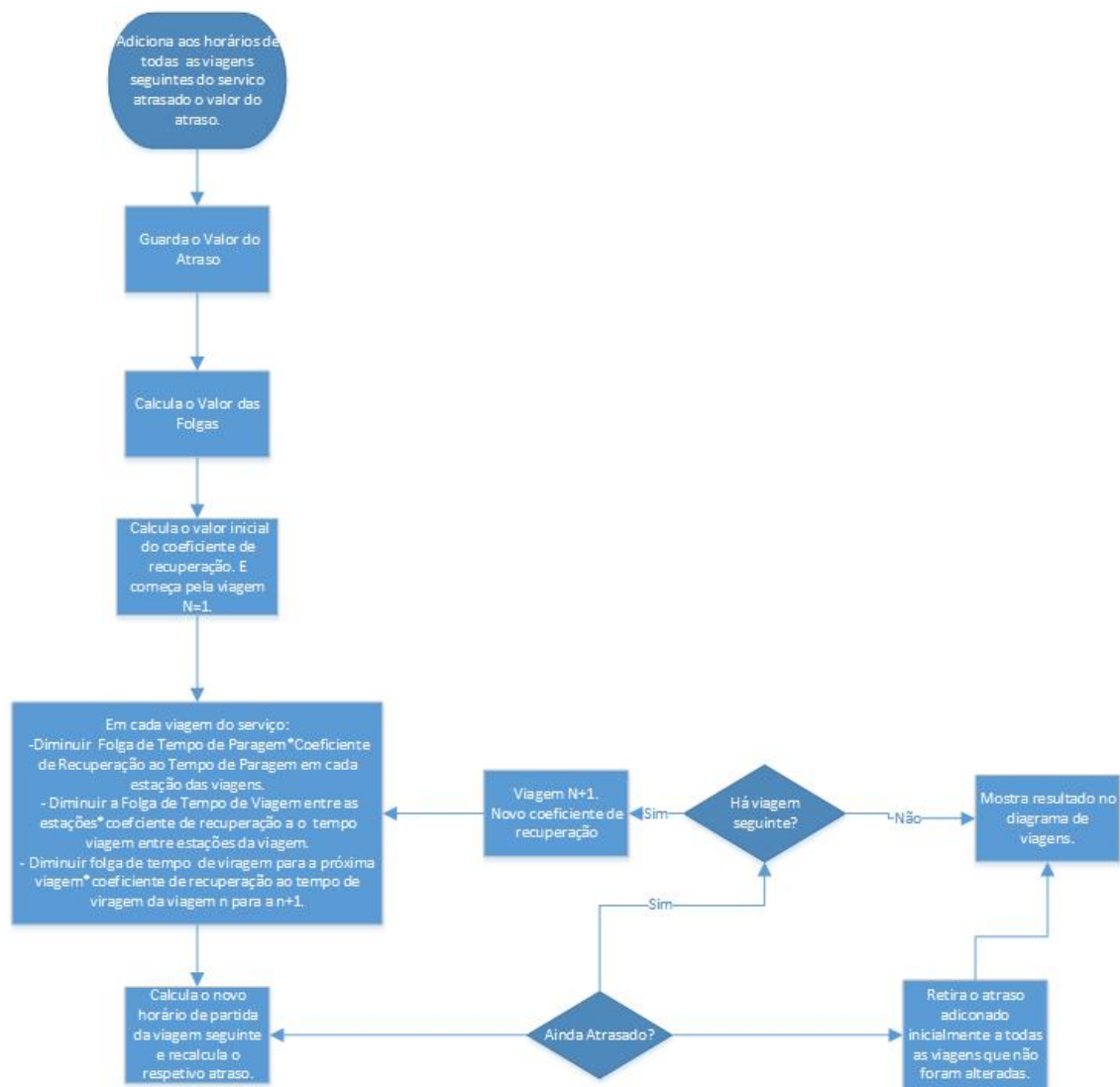


Figura 5.3: Algoritmo 1

5.3.3 Algoritmo 2

Este segundo algoritmo tem como objetivo dar uma solução de horário quando o atraso é considerado um atraso do Tipo 2. Este algoritmo terá como meta gerar sempre uma solução sem conflitos, para tal irá proceder de uma forma semelhante ao algoritmo 1. Mas, irá tratar todos os serviços dos horários, em vez de tratar apenas o serviço com atraso. Desta forma irá surgir um horário novo para todos os serviços, mesmo aqueles que poderiam não ser afetados diretamente pelo o atraso.

O algoritmo é constituído pelos seguintes passos:

0. Soma a todas as viagens de todos os serviços que seguem atrás a viagem onde ocorreu o atraso, e que ainda não se encontram na linha, o valor do atraso.

1. Ordena os horários por ordem de partida nos terminais.

2. Calcula o coeficiente de recuperação inicial.

3. Com o horário ordenado começa na primeira viagem do serviço e vai diminuir os tempos de paragem, tempos de viagem e tempos de viragem dessa viagem, tendo em conta o coeficiente de recuperação e também as folgas.

4. Em seguida, em vez de continuar o procedimento na viagem seguinte do serviço, irá passar para a primeira viagem do serviço seguinte.

5. O passo 2 e 3 são repetidos até as primeiras viagens de todos os serviços terem sido tratadas.

6. Em seguida calcula-se o novo atraso. Se o valor for maior que zero voltamos ao passo 2, na viagem seguinte à alterada.

7. Estes passos são repetidos até o atraso ter sido recuperado ou então quando não houver viagens.

8. Por fim é mostrada a solução no diagrama de viagens.

Estes serão os passos base do algoritmo. Na imagem [5.4](#) apresentamos o sua descrição mais detalhada.

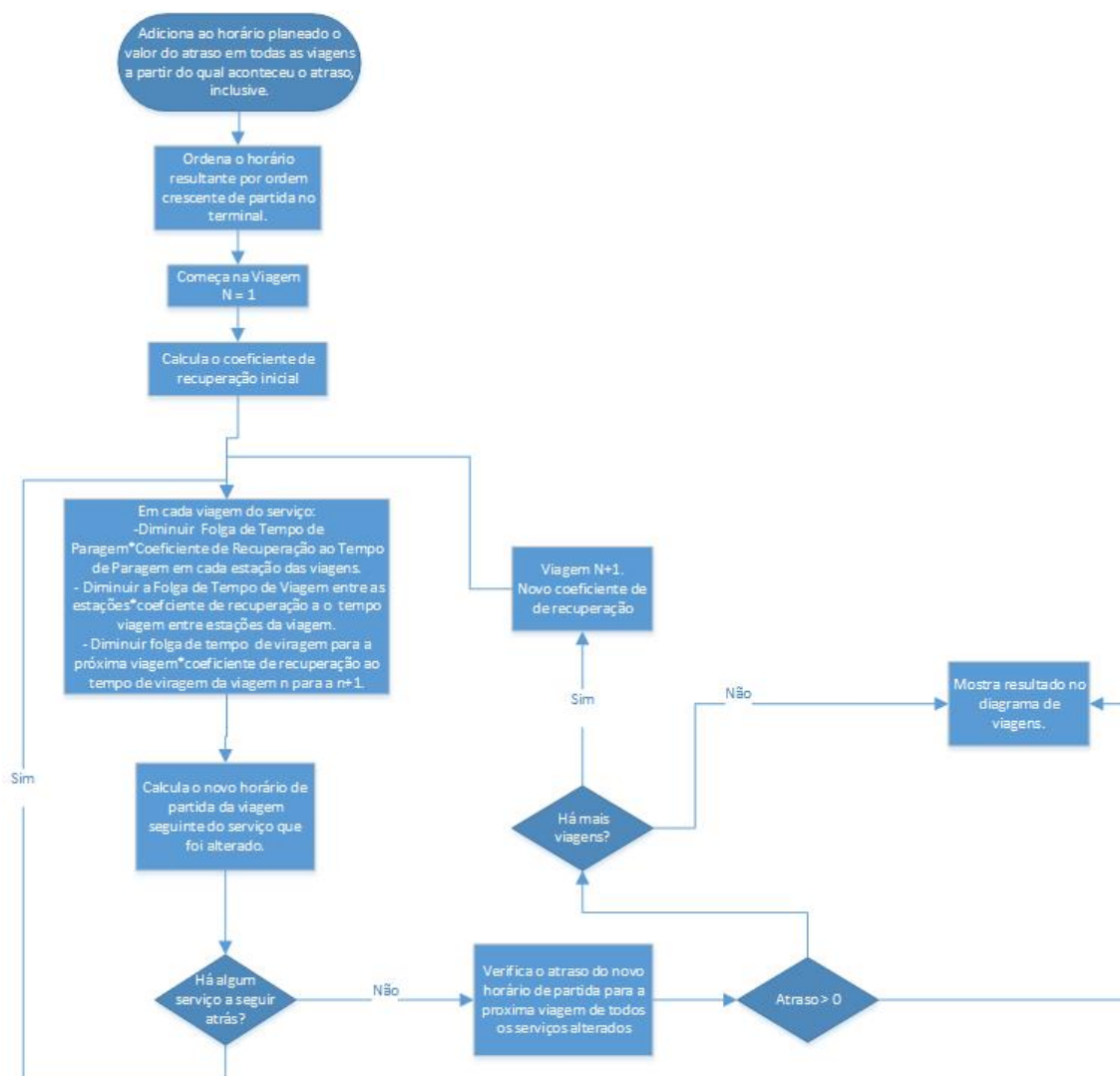


Figura 5.4: Algoritmo 2

Capítulo 6

O protótipo

Este capítulo tem como objetivo mostrar como foi implementado o procedimento descrito anteriormente e explicar todas as razões que levaram a este tipo de implementação. Não é apenas objetivo do trabalho validar os algoritmos. Também em relação à tecnologia há o objetivo de perceber as vantagens e as desvantagens da abordagem utilizada. Num primeiro momento iremos mostrar as razões que levaram à escolha das tecnologias utilizada, e serão apresentadas as mesmas. Num segundo momento é feita uma descrição de todo o interface, e sua forma de funcionamento. Por fim, iremos apresentar os resultados obtidos pela aplicação.

6.1 Descrição geral

6.1.1 Introdução

Nos dois capítulos anteriores foi apresentada exaustivamente a dinâmica do problema e todos os algoritmos propostos para implementar uma manobra de regulação que tem como objetivo recuperar os horários de um atraso: o deslizamento. Mas foi, então, preciso verificar os resultados dos mesmos. Haveria então várias possibilidades para o fazer: poderíamos tentar utilizar o software já existente na EFACEC, mas por razões de integração e alteração no código já existente esta opção foi descartada; poderíamos optar pela implementação de um modelo de regulação em ferramentas como o Excel ou MatLab, opção mais simples devido às enormes bibliotecas e APIs já existentes neste tipo de pacotes de software mas tal foi descartada pois como foi dito antes, era objetivo de estudo perceber alternativas à forma e às tecnologias utilizadas pela empresa e este tipo de ferramentas não são alternativas viáveis de comercialização; poderíamos então desenvolver de raiz um software que tivesse as funcionalidades básicas necessárias para poder implementar a algoritmia anterior e foi o que realmente se desenvolveu.

Na secção 3.3.3 foi sugerido alguns pontos de melhoria na ferramenta da empresa, tal como esta poder estar um pouco obsoleta em relação ao interface como também o seu modelo de desenvolvimento, foi então com os requisitos de desenvolver algo que:

- permitisse um desenvolvimento de interfaces mais moderno e atualizados;

- seguisse padrões de desenvolvimento, que garantisse uma harmonia para a divisão do trabalho em equipa (não nos podemos esquecer da complexidade desta ferramenta e do número de pessoas que a desenvolvem);
- executasse em ambiente *Windows*.

É importante destacar que todo o desenvolvimento deste protótipo tentou ser o mais semelhante do sistema real existente na empresa, para tal foi também criada uma base de dados em *MySQL*, onde se encontram os dados referentes aos horários e características da linha, tal como o sistema real.

6.1.2 As linguagens e os ambientes de desenvolvimento

No desenvolvimento de software é elemento chave a escolha da linguagem a utilizar, como também o IDE (*Integrated development environment*). No entanto, não é tarefa fácil, pois o número de linguagens existentes é na ordem dos milhares e tende a vir a aumentar com o passar do tempo. Em grande parte das empresas de software, a linguagem é escolhida grande parte das vezes com base em dois critérios, os requisitos do projeto, e mais importante, a *expertise* que possui na linguagem, sendo este último fator por vezes limitador no aumento da qualidade ou aparecimento soluções inovadoras. A abordagem para a escolha destes dois elementos tentou romper com este paradigma. A escolha centrou-se exclusivamente nas suas potencialidades e nos requisitos da mesma.

Em termos técnicos os grandes requisitos da implementação são: possibilidade de desenhar um interface moderno e inovador e que tivesse a capacidade de tratar uma quantidade de dados considerável. Com estes requisitos decidimos destacar duas linguagens, que no nosso entender são as mais importantes neste tipo de contexto, Java e C#, por serem linguagens muito completas, de alto nível e com enorme suporte para criação de interfaces.

Entre as duas, a escolha recaiu no C# pois foi tido em conta que o objetivo era desenvolver algo para correr exclusivamente em Windows, e muito também pelo o excelente IDE que o *Windows* tem para desenvolver aplicações nesta linguagem, o *Microsoft Visual Studio*. Considerado por muitos como um dos melhores IDE para desenvolvimento de software, possuindo inúmeras ferramentas para uma maior eficiência do trabalho em equipa, ferramentas de análise, ferramentas de supervisão e ferramentas de teste, elementos que quando bem utilizados poderão ser fator diferenciador na engenharia de software e no sucesso do projeto.

Depois de escolhida a linguagem ainda terá que ser escolhida qual será a base para a produção do interface, pois haverá várias formas para o fazer. Dentro da linguagem escolhida e assente na Framework .NET temos várias opções, mas as que se poderão adequar ao projeto serão as *Windows Forms Application* e as WPF (*Windows Presentation Foundation*), pois não queremos uma aplicação tipo web. Depois de perceber que a WPF era uma versão mais recente das *Windows Form Application*, e que tem uma quase infinita capacidade para desenvolvimento de interfaces desde dos mais simples aos mais complicados como jogos em 3 dimensões. WPF é uma linguagem

declarativa, XAML, ao estilo HTML, onde é possível criar para o mesmo interface diferentes estilos, análogo ao HTML e CSS.

Não é o propósito deste projeto fazer uma avaliação às linguagens, nem uma exaustiva análise sobre as possibilidades de desenvolvimento. No entanto, é importante perceber quais foram as razões destas escolhas.

6.1.3 Model View ViewModel

No meio profissional, o desenvolvimento de software segue normalmente um processo bem definido, conhecido em inglês como, software *development life-cycle*. Existem vários modelos de desenvolvimento, tais como, *Agile*, *Waterfall*, *Spiral*, *Iterative* e *Incremental Model*, entre muitos outros. Este processo é normalmente bem documentado, e depende do tipo de software a desenvolver, tal como acontece na EFACEC.

No entanto, na fase de implementação, existem inúmeros factores que são tidos em conta pelos programadores, tais como: design de interfaces, interacções de interface, conectividades, segurança, multithreading, validação, testes, etc. E, se não for seguida uma metodologia, o desenvolvimento poderá ter resultados desastrosos, e até provocar o insucesso do projeto. Por isso é importante, seguir uma metodologia, e no campo da engenharia de software dá-se o nome de design patterns, e existem em grande número dependendo do objetivo do software, e das linguagens escolhidas para desenvolver. É de referir, que muitas vezes os programadores utilizam as suas próprias metodologias, não querendo avaliar quais serão melhores, podemos sim afirmar que em trabalho equipa será mais fácil optar por uma metodologia padronizada.

No desenvolvimento de aplicações WPF, podemos encontrar conhecidas metodologias, tais como *Model View Controller* (MVC), *Model View Presenter* (MVP), e *Model View ViewModel* (MVVM). Embora a escolha tenha recaído na metodologia MVVM, por ser ter tido a sua origem na *Microsoft*, por encaixar bem para desenvolvimento de aplicações suportadas por eventos, as diferenças entre estas são algo ténues e que pouco interesse tem para o projeto.

MVVM é uma metodologia que se deve aplicar quando se quer fazer um desenvolvimento de interface completamente separado pela lógica de negócio, também conhecido por Model. Esta metodologia tem como objetivo fazer uso da função de “data binding” no WPF, que remove virtualmente todo o código da camada de interface. Esta separação permite que design do interface seja desenvolvido de forma completamente separado do resto da aplicação, sem se preocupar com qualquer tipo de lógica aplicacional. Além desta vantagem também permite trabalho paralelo, de grande importância em trabalho de equipa. O diagrama da figura 6.1 representa a arquitectura desta metodologia, de forma simples cada elemento tem as seguintes propriedades [13]:

- *Model* – ou também conhecido como *data access layer* representa o estado real do conteúdo.
- *ViewModel* – é uma abstracção do view, e serve como mediador entre o *Model* e a *View*. Normalmente a informação é exposta através de propriedades e comandos à camada *View*, através de funcionalidade de *binding* do WPF.

- *View* – representa todos os elementos que são mostrados no ecrã (botões, labels e outros controlos)

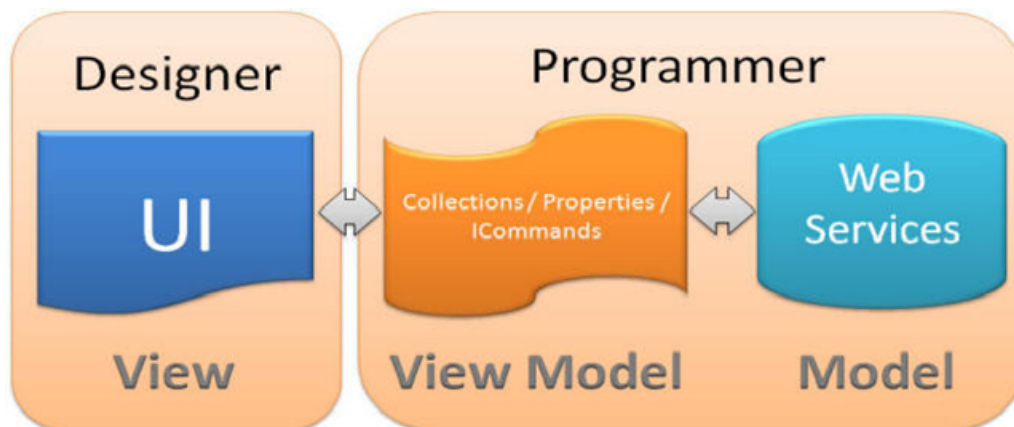


Figura 6.1: Arquitetura geral do padrão de desenvolvimento *Model View ViewModel* [2]

Por fim é importante referir que além das vantagens referidas, a utilização deste padrão de desenvolvimento trás grandes vantagens na fase de testes, permitindo a criação de rotinas de teste de forma muito mais simples e objetiva.

6.2 Descrição do interface

Na secção 3.3.3 foram referidas algumas limitações da plataforma, ou melhor, foram sugeridas algumas melhorias que esta poderia apresentar. O desenvolvimento deste protótipo tentou seguir essas ideias de forma a perceber se realmente estas teriam resultado e ainda tentou seguir, mesmo que de forma simplificada a arquitetura real deste tipo de aplicação, tendo sido por isso criado uma base de dados onde estariam os horários e onde estes se atualizariam em tempo real.

Na figura 6.2, podemos ver o resultado final. Como podemos observar o interface está dividido em três grandes áreas. Na área 1, estão duas tabelas, onde estão representados os horários de cada viagem de forma detalhada, sendo que a cada tabela representa um sentido. Na área 2, temos um diagrama de viagens, onde é mostrado de forma gráfica os horários planeados, também será nesta zona do interface que serão apresentados os resultados de cada simulação. Este gráfico também possui a possibilidade de zoom através da barra de scroll, permitindo ao operador navegar na solução e visualizar as linhas de forma mais pormenorizada. Na área 3, é onde estão as manobras de regulação, neste caso concreto de prototipagem apenas a manobra de deslizamento está implementada.

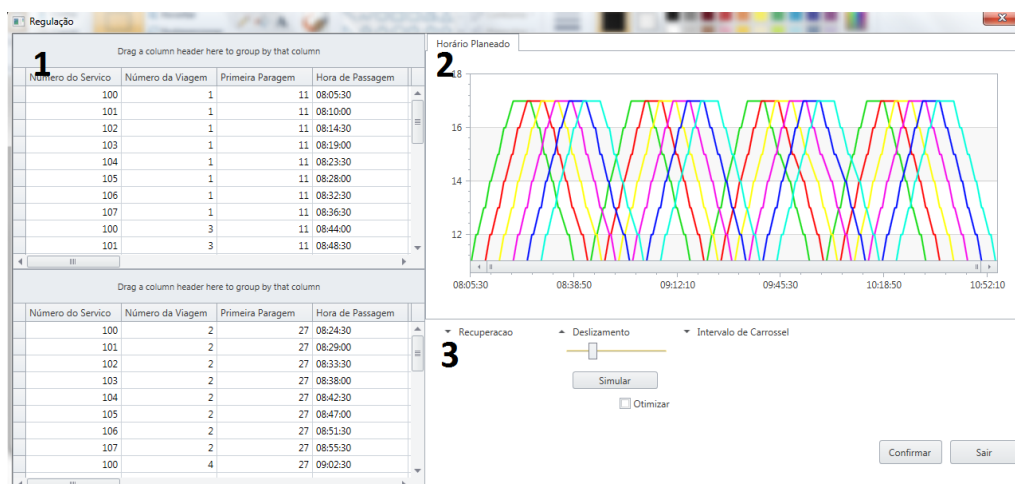


Figura 6.2: Aspeto geral do interface do protótipo desenvolvido

Agora de forma mais detalhada iremos apresentar o modo de funcionamento da aplicação como também todas as suas funcionalidades. As tabelas onde estão os horários como também os respetivos atrasos, são completamente customizáveis e permitem uma navegação bastante flexível. Como vemos na figura 6.3 é possível organizar a tabela por qualquer coluna, ou múltiplas colunas. Na figura podemos verificar o resultado quando organizada pelos número de serviço, para o fazer basta arrastar a coluna para o cabeçalho da tabela. Esta funcionalidade tem por base uma biblioteca de controlos para aplicações WPF da DevExpress.

Estas tabelas deverão ser atualizadas de forma automática com uma frequência de tempo definida, recebendo o atraso com que os veículos estão a chegar às estações. A partir destes dados o operador poderá tomar alguma decisão. Se decidir que quer deslizar o atraso, deverá selecionar na tabela a viagem atrasada, de seguida clicar em deslizamento, e se houver a intenção de diluir esse atraso nas viagens seguintes deverá ter a checkbox "otimizar" selecionada. Por fim deverá escolher a forma como quer ver esse atraso diluído, para tal deverá utilizar o *slider* disponível, sendo que se este slider estiver mais próximo da extremidade esquerda, o atraso será diluído em menos viagens, se o slider estiver mais próximo da extremidade direita, o atraso será diluído por mais viagens. Este slider implementa, o coeficiente de recuperação descrito na secção 5.3.1. Depois de seguir estes passos, pressiona-se simular, e surgirá na zona do diagrama de viagens um novo separador, com o resultado da simulação. O gráfico mostrará os horários planeados com linhas sólidas, e a tracejada apresenta o resultado da simulação. Poderão ser feitas múltiplas simulações e por fim se decidirmos submeter alguma das simulações, deveremos ter o separador correspondente aberto e pressionar submeter.

É importante referir que estamos perante uma versão de protótipo, tendo por isso havido alguns pormenores que não foram tidos em conta.

| Número do Serviço ▲ | | | | |
|--------------------------|------------------|------------------|----------|--|
| Número da Viagem | Primeira Paragem | Hora de Passagem | Paragem2 | |
| ▼ Número do Serviço: 100 | | | | |
| 1 | 11 | 08:05:30 | | |
| 3 | 11 | 08:44:00 | | |
| 5 | 11 | 09:21:30 | | |
| 7 | 11 | 10:00:00 | | |
| ▶ Número do Serviço: 101 | | | | |
| ▶ Número do Serviço: 102 | | | | |
| ▶ Número do Serviço: 103 | | | | |
| ▶ Número do Serviço: 104 | | | | |
| ▶ Número do Serviço: 105 | | | | |

Figura 6.3: Tabela de navegação dos horários

6.3 Resultados

Após ter sido apresentada a aplicação desenvolvida será essencial perceber se realmente esta responde da forma esperada, e para tal foram feitos alguns testes. Os testes desenvolvidos tiveram por base um horário fictício, com restrições fictícias, onde se forçou diferentes valores de atraso e se executou diversas simulações de deslizamento desse mesmo atraso. É importante relembrar que em todos os diagramas de viagem apresentados em seguida, as linhas sólidas são o horário planeado, e as tracejadas são o resultado do deslizamento, o objetivo é perceber a diferença do novo horário para o horário planeado, também de referir que serão apenas apresentados os diferentes diagramas resultantes da simulação em vez de todo o interface gráfico.

Cenário 1

No primeiro cenário de teste simulamos um atraso menor do que a diferença de tempo que

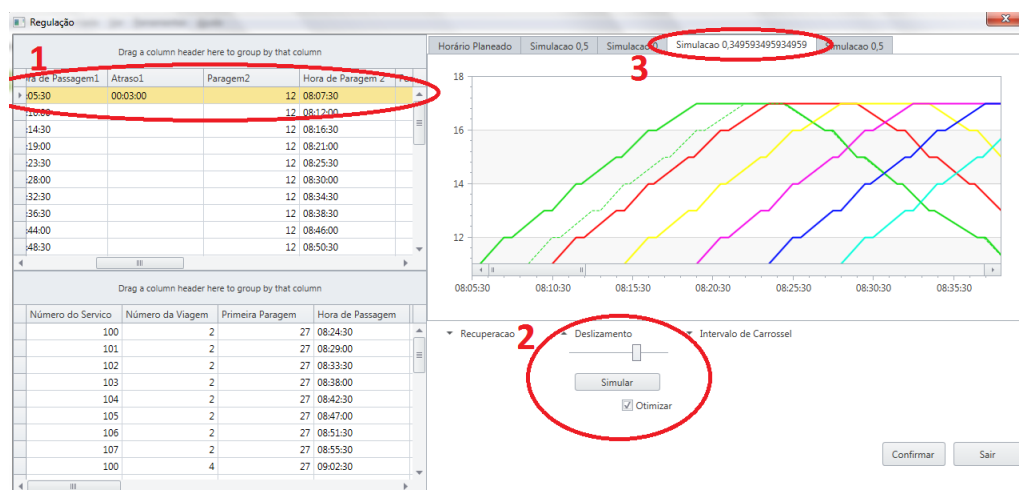


Figura 6.4: Figura ilustrativa da forma de funcionamento da aplicação

existe entre esse veículo e o que segue atrás. O atraso inserido tem o valor de 3 minutos, e foram realizadas 3 simulações com diferentes valores de coeficiente de recuperação, ou seja, diferentes formas de ver o atraso diluído nas viagens seguintes.

Na imagem 6.5 podemos ver o resultado da simulação, quando colocado o *slider* do coeficiente de recuperação todo chegado ao seu extremo direito, isso representa uma intenção de querer ver o atraso diluído de forma lenta ou em mais viagens. Como vemos na imagem, só existe uma linha a tracejado, pois como o atraso era do Tipo 1, apenas esse serviço foi alterado, também se pode observar que ao longo do tempo a linha a tracejado a verde tende a aproximar-se da linha verde sólida como era de esperar, e quando a linha a tracejado fica por cima da linha sólida, significa que o atraso foi totalmente diluído. Neste caso concreto podemos ver que só ao fim de quatro viagens é que o atraso foi totalmente recuperado.

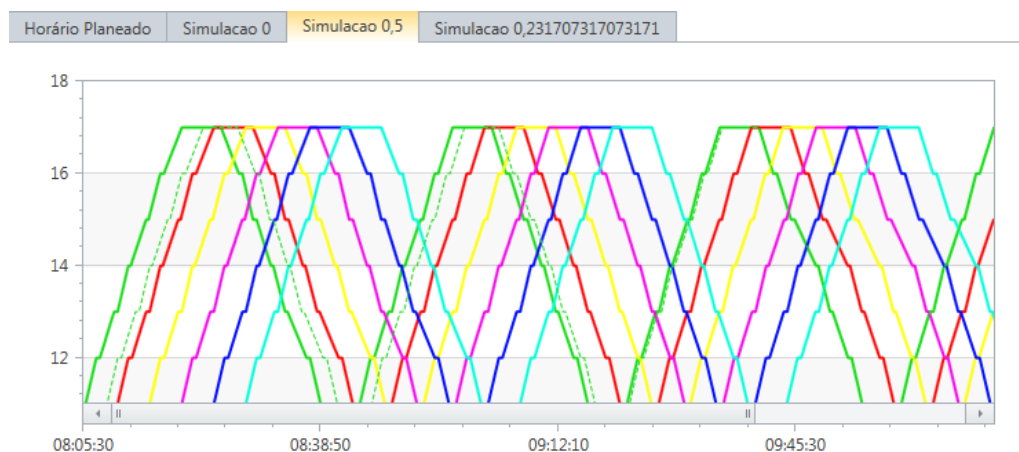


Figura 6.5: Resultado do deslizamento de um atraso de 3 minutos com coeficiente de recuperação mínimo

As figuras 6.6 e 6.7 mostram o resultado de duas outras simulações do mesmo atraso. Como é fácil perceber, a figura 6.6 é onde a recuperação do atraso é realizada de forma mais rápida, ao fim de 2 viagens já se encontra com o mesmo horário que tinha no horário planeado. Esta simulação foi feita colocando o slider do interface no sua extremidade esquerda, ou seja os tempos de folga que havia para gastar na viagens seguintes foram gastos na sua totalidade até o horário voltar a estar alinhado ao horário planeado. A imagem 6.7 mostra o cenário intermédio, tendo sido o *slider* colocado no meio, desta forma a solução obtida já foi distinta de todas as outras, conseguindo recuperar o atraso após três viagens.

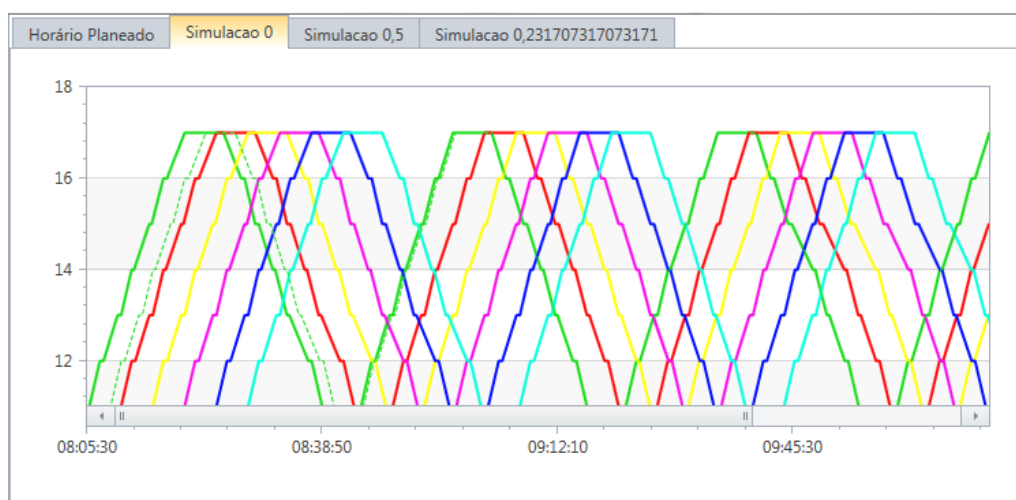


Figura 6.6: Resultado do deslizamento de um atraso de 3 minutos com coeficiente de recuperação máximo

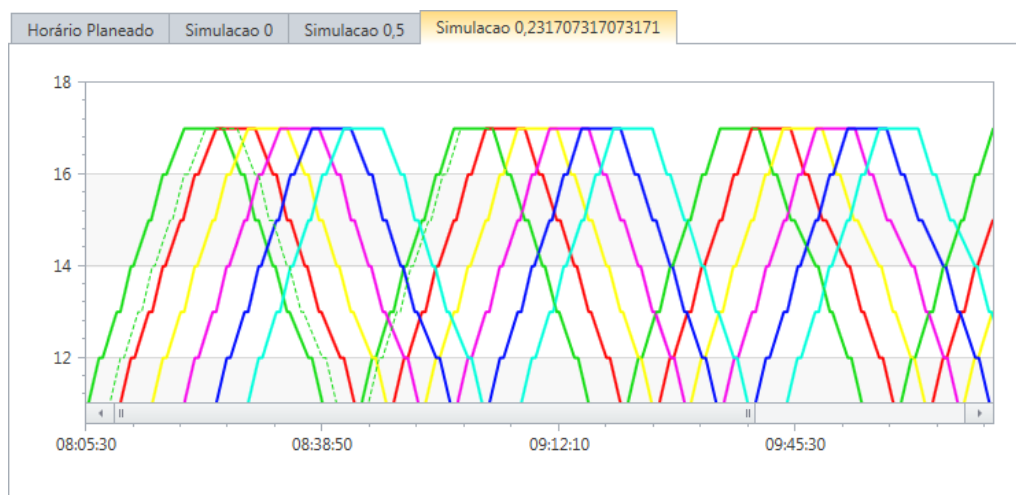


Figura 6.7: Resultado do deslizamento de um atraso de 3 minutos com coeficiente de recuperação médio

Os resultados deste primeiro cenário de testes foram o esperado, o software avaliou o atraso

a diluir, classificando-o de tipo 1, e foi posteriormente tratado então pelo algoritmo 1, 5.3.2. A partir da variação do slider foram criadas múltiplas soluções, podendo facilmente navegar entre elas e comparar qual a que poderá ser melhor para satisfazer as necessidades do utilizador.

Cenário 2

No segundo cenário, o objetivo é verificar o comportamento do sistema quando o atraso já é superior a diferença de tempo entre o veículo atrasado e o veículo que segue atrás, esta diferença será em relação ao seu horário planeado, pois como foi dito não haverá ultrapassagens neste problema. O atraso inserido tem um valor de 10 minutos. Em seguida serão mostrados dois resultados de duas simulações distintas, no entanto é importante desde já estar ciente de alguma dificuldade na interpretação da imagem, será importante apenas ter em atenção as linhas a tracejado, e a forma como elas se vão aproximando das linhas sólidas das suas respectivas cores. O interface possibilita uma navegação mais amigável do utilizador, pois permite ampliar o gráfico, mas para o contexto do documento é importante mostrar o resultado de uma forma global, para perceber as grandes vantagens de utilizar sistemas de apoio à decisão que permitam simulação.

Na imagem 6.8, vemos o resultado da simulação quando o slider é colocado na sua extremidade esquerda, ou seja, o atraso será recuperado o mais rapidamente possível. Como é fácil perceber, todos os veículos foram atrasados 10 minutos, mas ao fim de 2 viagens já se encontrarão alinhados com os seus horários planeados.

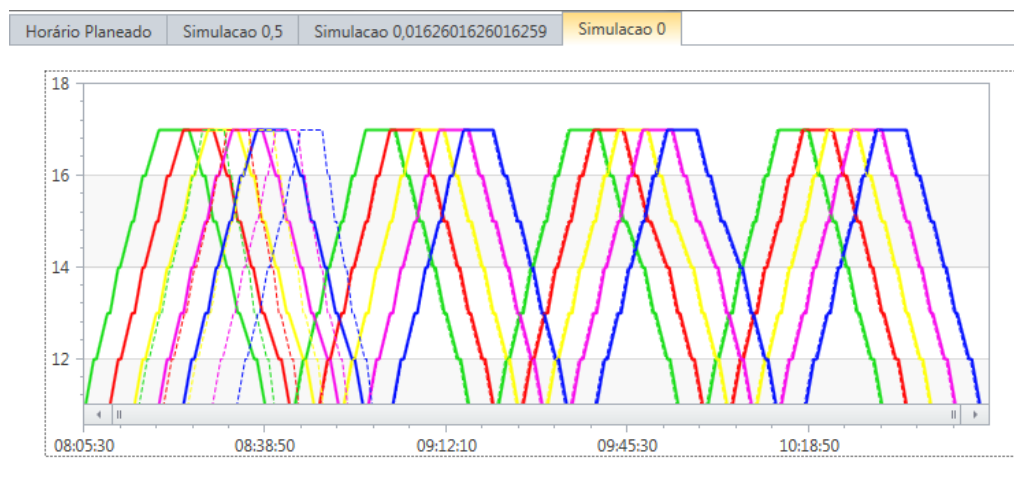


Figura 6.8: Resultado do deslizamento de um atraso de 10 minutos com coeficiente de recuperação máximo

Na imagem 6.9 é possível observar o resultado da simulação quando se quer um atraso recuperado de uma forma mais lenta possível. Neste caso podemos ver que ao fim das 8 viagens que existiam no horário, o atraso nunca foi recuperado na sua totalidade. A diferença entre esta imagem e a anterior é clara, não podendo dizer qual a melhor, pois esse é a grande vantagem do sistema, ele permite uma maior flexibilidade de soluções.

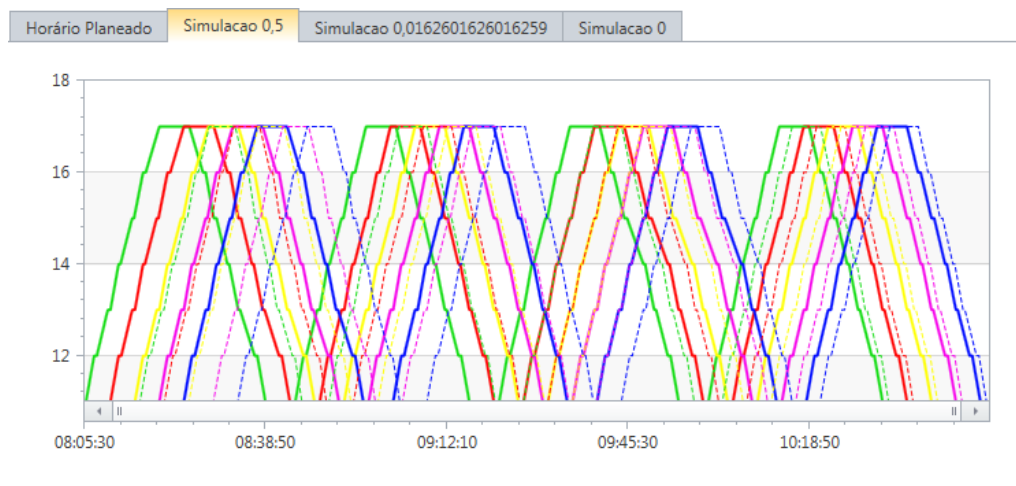


Figura 6.9: Resultado do deslizamento de um atraso de 10 minutos com coeficiente de recuperação mínimo

Como é natural, poderiam ter sido feitos mais testes com diferentes horários, poderíamos ter colocado janelas temporais mais alargadas, mas tornar-se-ia algo complicado de representar no documento, pois não haveria a possibilidade de navegação na solução.

Capítulo 7

Conclusões

7.1 Conclusões sobre o trabalho realizado

Este trabalho resultou de um projeto que teve uma duração aproximada de quatro meses, desenvolvido no grupo EFACEC, tendo como propósito o estudo de um sistema de apoio à regulação do tráfego metroferroviário.

Este estudo permitiu perceber a importância da existência de softwares que permitam que as pessoas que tem que gerir uma rede metroferroviária em tempo real, possam fazê-lo de forma o mais eficiente possível, resultando numa maior qualidade do serviço na perspetiva do utilizador. Este tipo de software desempenha um papel fundamental, pois a tarefa de supervisão e controlo de uma rede deste tipo envolve tantas variáveis e tem restrições temporais tão apertadas que será praticamente impossível para o humano poder fazê-lo de forma eficiente sem o apoio de uma plataforma computadorizada.

Simulação

Em relação à proposta de resolução há alguns pontos que se devem destacar. Em primeiro lugar, é de realçar a possibilidade de simular diferentes cenários. Esta opção permite tomar decisões mais acertadas, e também reduzir correções que poderiam ter que ser feitas quando não há possibilidade de simulação. A forma encontrada para dar *feedback* ao operador através do diagrama de viagens revelou-se bastante poderosa, pois permite uma visão global dos horários, algo muito complicado de obter através de uma tabela de horários. No entanto, será importante referir que mesmo através deste diagrama poderão surgir algumas dificuldades quando o número de serviços for grande, e por consequência haverá muitas linhas no gráfico a cruzarem-se tornando-se complicado fazer uma avaliação precisa da solução. Estas dificuldades tem tendência a diminuir quando o operador tem alguma experiência na interpretação deste tipo de ferramentas, este facto resulta de ter sido colocado o interface perante pessoas com experiência na área e outras sem qualquer tipo de conhecimento sobre a temática.

Coefficiente de recuperação

A forma encontrada para dar ao utilizar a opção de escolher como prefere que o atraso seja diluído resultou como esperado, de forma simplicíssima através de um *slider* o operador poderá criar

soluções distintas, soluções que mais se adequam às necessidades. Esta opção permite que surjam resultados com horários mais rígidos, ou seja com menor ou mesmo sem nenhuma possibilidade de recuperação nas próximas viagens, pois os tempos de folga foram todos gastos, ou permite dar uma solução que recupera o atraso de forma mais lenta, resultando num horário menos rígido e com possibilidade para novas tentativas de deslizamento nas viagens seguintes, se algo não correr como o esperado. O *slider* tem a grande vantagem de ser algo simples, bastante intuitivo mas não é algo preciso do ponto de vista do operador.

Tipos de atraso

A distinção dos atrasos em dois tipos, permitiu poupar esforço computacional pois quando se esta perante um atraso Tipo 1, só terá de ser alterado um serviço, ao invés de todos, o facto de também não termos que envolver todos os serviços também mantêm o horário mais fiel ao horário planeado. Mas aqui, poderão ser feitas algumas críticas, pois em relação ao deslizamento de um atraso do tipo 1, poderá resultar um horário, onde existam, pelo menos, nas viagens seguintes ao atraso, uma proximidade temporal entre o serviço que segue atrás do serviço atrasado, podendo não ser benéfico do ponto de vista da companhia de transportes que correrá o risco de ter um veículo sobrecarregado e o veículo que segue atrás por ter uma diferença de tempo curta, pode estar vazio.

Interface

A decisão de apresentar o horário em tabelas muito semelhante aos horários que estão disponíveis aos passageiros, havendo uma separação dos sentidos de circulação trouxe a vantagem de dar ao operador uma visão mais completa, e de fácil interpretação, podendo facilmente perceber quando é que o veículo deve estar em determinada estação, qual o atraso com que chegou, permitindo uma análise preditiva do ponto de vista do operador. Mas, também aqui, se percebeu que quando se esta perante uma linha com muitas estações e com maior complexidade, ou seja, troços comuns nos dois sentidos, múltiplas linhas interligadas, não será tão viável a existência de duas tabelas, pelo menos não desta forma.

Linguagem utilizada

Na fase de implementação propôs-se perceber a viabilidade da tecnologia utilizada e também a forma de desenvolvimento. É então, importante perceber quais as conclusões relações retiradas desta abordagem.

Em relação ao desenvolvimento de uma aplicação WPF (C# +XAML), terá que ser dito que é uma tecnologia extremamente poderosa, é uma ferramenta que tem grandes capacidades de desenvolvimento principalmente ao nível do interface, grande capacidades de separação entre o *Graphica User Interface* (GUI) e a lógica de negócio. Mas, esta ferramenta terá tanto de poderosa, como de complicada, esta tecnologia tem várias dificuldades: curva de aprendizagem lenta, inúmeras formas de fazer a mesma tarefa, sendo complicado perceber qual a melhor.

Modelo de desenvolvimento

Em relação ao modelo de desenvolvimento utilizado, o MVVM, tem que ser dito que encaixa de forma quase perfeita na tecnologia utilizada, o desenvolvimento utilizando este modelo

facilitará o trabalho de equipa, pois possibilita partir a aplicação em *layers*. No entanto, a implementação deste modelo tem algumas variantes, será então importante seguir uma *framework* como por exemplo *PRISM*, ou *Caliburn*. Durante desenvolvimento também se percebeu que este padrão de desenvolvimento é pouco produtivo quando se trabalha de forma individual e quando o objetivo é implementar algo simples.

Por fim, poderia dizer que só valerá a pena utilizar este tipo de tecnologia e padrão de desenvolvimento, para o desenvolvimento de software complexo, em equipa, e onde as pessoas tenham bastante experiência com a ferramenta, ou então que estejam dispostas a perder alguns meses para se adaptarem à ferramenta. Se não for este o caso deverão procurar formas alternativas, como por exemplo as Windows Forms Applications, ou então ferramentas Web(HTML5, JavaScript).

7.2 Conclusões sobre a experiência empresarial

A elaboração desta tese de mestrado numa empresa, mais concretamente no Grupo EFACEC, foi algo extremamente enriquecedor. A experiência de ter contacto com a realidade empresarial permitiu conhecer um produto real, e com uma enorme complexidade, envolvendo diferentes áreas do conhecimento, algo que na faculdade poucas vezes acontece.

Há uma palavra que caracterizará este projeto, autonomia. Todo este projeto, resultou do estudo feito da plataforma da empresa mas em nenhum momento fui obrigado ou sequer influenciado para qualquer tipo de abordagem. Durante o desenvolvimento foi-me dada total liberdade para escolher o rumo a dar ao problema, sendo que todas as análises e decisões tomadas foram pessoais.

Ao nível das competências técnicas, aprendi duas novas linguagens de programação de raiz, C# e XAML, melhorei os meus conhecimentos ao nível de Base de Dados e também adquiri um maior à vontade com problemas de *scheduling*.

Em relação a competências *soft*, também permitiu perceber ainda de forma mais acentuada a enorme importância do trabalho em equipa. Nos dias de hoje pouco ou nada se fará sozinho, e por isso é fator essencial saber trabalhar em equipa.

7.3 Trabalhos futuros

Como esperado, o principal resultado deste projeto foi o protótipo de uma aplicação informática para apoiar a decisão dos reguladores. Assim, há, naturalmente, alterações que poderiam ser feitas com ao melhoramento da aplicação. Como identificado 7.1, sugerimos as seguintes alterações:

- em relação ao diagrama de viagens, poderia ser acrescentado uma opção para configurar o que queremos ou não mostrar no gráfico;
- ser colocada uma *TextBox* para o utilizador colocar em quantas viagens gostaria de ver o atraso recuperado;

- o utilizador ter a possibilidade de forçar a que um atraso do Tipo 1 pudesse ser tratado com um atraso do Tipo2;
- melhorar a tabela onde são apresentados os horários ao utilizador. No caso de existirem várias linhas, poderia ser colocada uma *listbox* onde se poderia seleccionar quais os horários que queremos visualizar.

Os resultados obtidos do caso de estudo foram satisfatórios. No entanto se realmente se quiser seguir com o desenvolvimento desta aplicação para ser comercializada seria preciso ainda fazer alguns testes. Seria importante fazer testes com dados reais, e teriam de ser inserido algum detalhe, principalmente em relação a algumas especificidades de determinadas linhas. Por fim, também seria necessário completar a aplicação com mais algumas manobras de regulação.

Referências

- [1] EFACEC. Efacec. http://www.efacec.pt/presentationLayer/efacec_home_00.aspx?idioma=1, MONTH=March, YEAR=2013.
- [2] Abram John. Using model-view-viewmodel (mvvm) pattern in managing your wpf/silverlight application. <http://abramlimpin.wordpress.com/2011/02/03/using-model-view-viewmodel-mvvm-pattern-in-managing-your-wpfsilverlight-application/>, MONTH=February, YEAR=2011.
- [3] Karim Bouamrane, Christian Tahon, Marc Sevaux, e Bouziane Beldjilali. Decision making system for regulation of a bimodal urban transportation network, associating a classical and a multi-agent approaches. *Informaticas*, Vol. 16, No.4, páginas 473–502, January 2005.
- [4] Pierre Borne, Bisma Fayeche, Slim Hammadi, e Salah Maouche. Decision support system for urban transportation networks. *IEEE Transactions on systems, man, and cybernetics-part c: applications and reviews*, vol.33,no. 1, páginas 67–77, February 2003.
- [5] Pedro Alexandre de Carvalho Afonso. Railway traffic management. Relatório técnico, Instituto Superior Técnico - Universidade Técnica de Lisboa, September 2008.
- [6] Farzad Torkamani, Samira Fallah, e Masood Saadatmand. How urban managers can use dss to facilitate decision making process: An application of fuzzy topsis, February 2012.
- [7] Frada Burstein e Clyde W. Holsapple. *Handbook on Decision Support Systems 1: Basic Themes*. Springer., 2008.
- [8] Mahomed Mahomoud Ould Sidi. Contribution à l'amélioration des systèmes d'aide à la décision pour la régulation du trafic dans les réseaux de transport collectif. Relatório técnico, Université des Sciences et Technologies de Lille, December 2006.
- [9] Constance Paquel. Metro traffic regulation and quality of passenger service: performance analysis of the operation control centre at ratp. Relatório técnico, Delft University of Technology, July 2011.
- [10] Mohamed Mahmoud Ould Sidi, Slim Hammadi, Said Hayat, e Pierre Borne. Urban transport network regulation and evaluation: A fuzzy evolutionary approach. *IEEE Transactions on systems, man, and cybernetics-part c: applications and reviews*, vol.38,no. 2, March 2008.
- [11] Flavien Balbo e Suzanne Pinson. Using intelligent agents for transportation regulation support system design. *Transportation Research Part C: Emerging Technologies*, páginas 140–156, 2010.

- [12] Selim Dunder e Ismail Sahin. Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways. *Transportation Research Part C: Emerging Technologies*, páginas 1–15, November.
- [13] Josh Smith. Wpf apps with the model-view-viewmodel design pattern. <http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>, MONTH=February, YEAR=2009.